TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

ANTTI NEVAVUORI

DEVELOPMENT OF AN INTELLIGENT DATA COLLECTION INSTRUMENT FOR MOBILE EQUIPMENT

Master of Science thesis

Examiner: prof. Jouni Mattila
Examiner and topic approved by the Faculty Council of the Faculty of Engineering Sciences
on 8th April 2015

ABSTRACT

ANTTI NEVAVUORI: Development of an Intelligent Data Collection Instrument for Mobile Equipment
Tampere University of Technology
Master of Science Thesis, 62 pages, 6 Appendix pages
April 2016
Master's Degree Programme in Mechanical Engineering
Major: Machine Automation
Examiner: Professor Jouni Mattila

Keywords: Internet of Things, Industrial Internet of Things, data collection, COSMOS, vibration analysis

The Internet of Things (IoT) is the extended version of the Internet where physical "things" are connected to the Internet network. When these physical "things" can sense the environment and communicate with people and other "things", it can be created, for example, intelligent and real-time data collection and analyzation solutions.

The IoT solutions have already started to be present in consumer markets, but in the technology industry, these solutions have only recently become a topic of greater interest. Keskustekniikka has been involved in the development of the Industrial Internet of Things (IIoT) by developing the COSMOS product family. The COSMOS products activate the IIoT and provide solutions from a single application to a wider overall solution.

This thesis investigates how the COSMOS data collection system can be adapted to Sandvik's mining machines, including different types of drill rigs, loading and hauling machines, and crushers. The thesis introduces a product concept for an intelligent vibration-based data collection instrument. The instrument analyzes the vibration of a machine and uses the analyzation results for recording the operation times of the most common operation modes. The instrument must be easy to install on all types of machines and it should work with as little configuration as possible.

The functionality of the designed product concept was tested with two different types of drill rigs. The test results show the potentiality of the instrument; although there are clear errors in the measurement results, it can be assumed that with a little optimization, the results would be better. Also, it has to be noted that as a part of COSMOS product family, the instrument is not intended only for Sandvik's machines. There might be several other potential target customers whose machines may be more suitable for vibration analysis. This means that the tested software might work with those machines without additional development. Based on these facts, it can be stated that the product concept is worth further development.

TIIVISTELMÄ

ANTTI NEVAVUORI: Älykkään tiedonkeruuinstrumentin kehitys liikkuviin työkoneisiin
Tampereen teknillinen yliopisto
Diplomityö, 62 sivua, 6 liitesivua
Huhtikuu 2016
Konetekniikan diplomi-insinöörin tutkinto-ohjelma
Pääaine: Hydrauliikka ja automatiikka, koneautomaatio
Tarkastaja: professori Jouni Mattila

Avainsanat: esineiden internet, teollinen internet, tiedonkeruu, COSMOS, tärinäanalyysi

Esineiden internet (engl. Internet of Things, IoT) on laajaa huomiota herättänyt internetin laajennos, jossa fyysiset ”esineet” ovat liitetty osaksi kaikkien tuntemaan internetverkkoa. Kun nämä fyysiset ”esineet” pystyvät aistimaan ympäristöä ja kommunikoimaan ihmisten sekä toisten ”esineiden” kanssa, on mahdollista luoda muun muassa älykkäitä ja reaaliaikaisia tiedonkeruu- ja analysointipalveluita.

Kuluttajamarkkinoilla IoT-sovellukset ovat jo yleistymässä, mutta teollisuudessa nämä sovellukset ovat vasta viime aikoina herättäneet suurempaa kiinnostusta. Keskustekniikka on osaltaan ollut tuomassa esineiden internetiä teollisuuteen tarjoamalla teollisen internetin ja pilvipalvelut aktivoivan COSMOS-tuoteperheen. COSMOS-tuotteet tarjoavat ratkaisuja yksittäisistä sovelluksista aina laajempiin kokonaisratkaisuihin.

Tässä työssä tutkitaan kuinka COSMOS-tiedonkeruujärjestelmä voidaan liittää Sandvikin kaivoskoneisiin, joita ovat muun muassa erilaiset poralaitteet, lastaus- ja kuljetuskoneet sekä murskaimet. Tätä varten työssä suunnitellaan tuotekonsepti älykkäälle tärinäpohjaiselle tiedonkeruuinstrumentille, joka tärinää analysoimalla mittaa koneen yleisimpien operointitilojen operointiajat. Instrumentin tulee olla helposti asennettavissa mihin tahansa laitteeseen ja sen tulee toimia mahdollisimman pienellä konfiguroinnilla.

Suunnitellun tuotekonseptin toimivuutta Sandvikin laitteisiin testattiin kahdella eri poralaitteella. Testit osoittivat instrumentin potentiaalin; vaikka saaduissa tuloksissa esiintyi monia selkeitä virheitä, voidaan olettaa, että ohjelmistoa optimoimalla tuloksista saadaan tarkempia. Lisäksi on muistettava, että osana COSMOS-tuoteperhettä, instrumenttia ei ole tarkoitettu vain Sandvikin käyttöön. Markkinoilta saattaa löytyä muita mahdollisia asiakkaita, joiden koneissa jo testattu ohjelmisto toimisi riittävän tarkasti. Näiden tutkimuksessa esiin tulleiden faktojen pohjalta voidaan todeta, että tuotekonseptin kehitystä kannattaa jatkaa.

PREFACE

This thesis was conducted in co-operation with Keskustekniikka and Sandvik Mining and Construction. Keskustekniikka provided all the needed materials, hardware components, and technical support for the study. Sandvik offered the needed test environment and test machines. The examiner for the thesis was Professor Jouni Mattila from Tampere University of Technology.

I would like to thank all parties for making this project possible. Without the technical support and software solutions given by Mr. Timo Hanhimäki and Mr. Timo Kivimaa from Keskustekniikka, the project would not have been possible. From Sandvik, I would like to thank all the people who made the performed tests possible. Especially, thanks to Mr. Ville Svensberg for all the advice and support he has given me during the project. Also, great thanks to Professor Jouni Mattila for the guidance.

I would also like to thank Ms. Johanna Hämäläinen for helping me with grammatical issues. The greatest thanks go to my family who has supported me throughout my studies.

Tampere, 22.3.2016

Antti Nevavuori

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ARM	Architectural Reference Model
R	Sandvik's Requirement ID
DCS	Distributed Control System
DHCP	Dynamic Host Configuration Protocol
e.g	exempli gratia
ed.	editor or edition
et al.	Latin et alii or et alieae, and others
FC	Functional Component
FFT	Fast Fourier Transform
FG	Functionality Group
FP7	European Framework Programme 7
HTTPS	Hypertext Transfer Protocol Secure
ICT	Information and Communication Technology
ID	Identification
IIC	Industrial Internet Consortium
IIoT	Industrial Internet of Things
IoT	Internet of Things
IoT-A	Internet of Things Architecture
ISO	International Organization of Standardization
ITU	International Telecommunication Unit
LAN	Local Area Network
M2M	Machine-to-Machine communication
MAC	Media Access Control
MIT	Massachusetts Institute of Technology
OSI	Open System Interconnection
OVP	Overvoltage Protection
PCB	Printed Circuit Board
PLC	Programmable Logic Controller
RFID	Radio Frequency Identification
SMD	Surface Mount Device
TUT	Tampere University of Technology
UML	Unified Modeling Language
WAN	Wide Area Network
WLAN	Wireless Local Area Network

1. INTRODUCTION

The Internet of Things (IoT) is currently a widely discussed topic in the technology industry. The term is used to describe an extended version of the Internet where embedded devices, services, and people are able to interact with each other. The IoT solutions have already started become present in the consumer life, but in the technology industry, the IoT has just recently become a topic of greater interest. When machines, automation systems, and humans can monitor and interact with each other, it is possible to improve technological reliability, sustainability, and efficiency. For these features, it has been predicted that the IoT will be the next game-changing phenomenon in the industrial companies. [13, p. 9-22]

For the designing of the IoT solutions, several projects have tried to create a blueprint for the IoT systems. One example of this type of project is the Internet of Things Architecture (IoT-A) project, which created an Architectural Reference Model (ARM) for the IoT systems. The aim of this model is to decrease the diversity of technologies and protocols, providing a common foundation for building an interoperable IoT system. [16]

Keskustekniikka has become involved in the IoT trend by developing their own IoT product family – COSMOS. The COSMOS products enable the Industrial Internet of Things (IIoT) and provide overall solutions from sensors to a cloud. The COSMOS products can be used, for example, for real-time process monitoring or for preventive maintenance that reduces unnecessary maintenance and increases efficiency. This kind of real-time process monitoring system collects data from the equipment's bus or from its own sensors, after which it transfers the data to the main database. There the data is analyzed and visualized for the user. From the real-time process monitor the user can see the history and the current status of an equipment, possible alarms and warnings, and longtime reports. [17; 18]

This thesis investigates ways to adapt the COSMOS system to be used when using Sandvik's mining equipment. The system should provide a lighter option for Sandvik's existing data collection and monitoring systems and it should record the basic operation times of a mining machine, such as drilling and engine hours. The system should also be easy to install and should not need any machine-specific configuration. In addition, the system should work in all machine types of Sandvik. However, this thesis focuses only on couple type of drills, leaving the other machine types for further development.

To meet aforementioned requirements, this thesis approaches the problem from the point of view that it is best to collect data with an intelligent data collection instrument that measures the physical phenomena happening in the equipment. This decision is also supported by a little market research, which shows that working sensor-based condition monitoring solutions, which measure the utilization rates and conditions of the machinery, are already existing [7]. On the basis of this information, it can be assumed that it is possible to detect the required operation modes by measuring some physical phenomenon, such as vibration.

The core point of this thesis is to design and study the product concept for that intelligent data collection instrument. The concept determines which features and components the instrument must contain and how the vibration analysis could be used for the operation mode detection. Thanks to this design and study, Keskustekniikka extends their product family with this new product concept and Sandvik receives information about how a vibration-based data collection instrument works with their machines.

The thesis consists of six chapters. Chapter 2 gives a background information about the development process by introducing the IoT and the COSMOS system. Chapter 3 presents Architectural Reference Model (ARM) that helps to design IoT systems. Chapter 4 describes the development process of the product concept and represents the architectural models. Chapter 5 evaluates the successfulness of the product concept and Chapter 6 concludes the thesis.

2. INTERNET OF THINGS

The Internet has changed the world over the last few decades. Many of people’s everyday tasks, such as information retrieval, shopping, and banking have moved to the Internet, which has changed people’s lives. However, the next evolutionary step of the Internet has already begun. In this step, real world physical objects will also be connected, which brings the Internet to the next level. When these embedded physical objects are able to interact with each other, with people, and with services, the application opportunities are unlimited. This extended version of the Internet is known as the Internet of Things (IoT). [13, p. 9-22]

This chapter introduces the IoT: The first part talks about the definition of the IoT and compares it with Machine-to-Machine communication (M2M). The second and third part discuss the history of the IoT and present some factors that have led to the development of it. The last part introduces Keskustekniikka’s COSMOS product family that is an IoT-based data collection system and the key point of this thesis.

2.1 Definition

The term “Internet of Things” is a broad term, and for that reason, the definition might vary depending on the different sources used. At first, before the IoT, this thesis defines the term “Machine-to-Machine Communication” that is a base and an integral part of the IoT. These two terms are used interchangeably in many sources but as Höller et al., this thesis keeps them separated. [13, p. 9-22]

2.1.1 Machine-to-Machine Communication (M2M)

The M2M solution is a closed network, where interesting assets are remote monitored or controlled. Höller et al. define it as follows: *“M2M refers to those solutions that allow communication between devices of the same type and a specific application, all via wired or wireless communication networks. [13, p. 11]”* This means that the solution is not only connection between two machines, but it can contain several machines that communicate with a server in which a specific application is running. However, because the M2M solutions work in a closed network, they do not share data with other systems or allow connection to the Internet. [13, p. 11-14]

A typical M2M system is composed of four components as shown in Figure 2.1. The first component is an M2M Device that performs an actual task. The task varies according to the system, but it can be a simple sensing task or a more complex multifunctional sensing and control task. The simplest M2M devices are, for example, sensors, but more complex devices also have some data processing capacity. [13, p. 11-14]

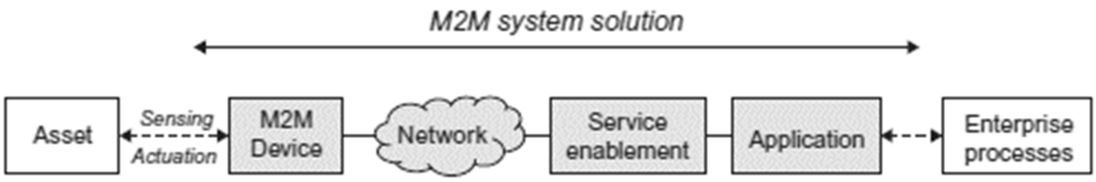

Figure 2.1. *Typical M2M solution [13, p. 12]*

The M2M Device is connected to a server via Network, which delivers information and commands of the system. The Network type is always selected according to the intended use. For example, the network can be a Bluetooth network, a Local Area Network (LAN), a Wide Area Network (WAN) or any other closed network. [13, p. 11-14]

A Service Enablement component provides generic functionalities that are common across a number of different applications. Hence, an application developer does not need to understand the complexity behind the functionalities and does not need to repeat the same job over and over again. The component is not necessary in an M2M system, but the use of it is becoming more popular because it reduces costs and eases application development. [13, p. 11-14]

The Application component creates the real business value for the owner. There are a number of different application types, such as a monitoring application that monitors and analyzes data received from the M2M Device. [13, p. 11-14]

Most M2M solutions are remote monitoring and control systems that are created to increase the efficiency, the safety, and the security of processes. The most common application areas in turn are telematics (e.g navigation systems), metering (e.g remote read electricity meters), remote monitoring (e.g monitoring of an interesting asset), fleet management (e.g vehicle positioning), and security (e.g surveillance camera systems). [13, p. 11-14]

2.1.2 Internet of Things (IoT)

Compared with the M2M, the IoT is a broader concept where the physical world merges with the digital world. Höller et al. define the IoT as follows: *"The IoT is a widely used term for a set of technologies, systems, and design principles associated with the emerging wave of Internet-connected things that are based on the physical environment. [13, p. 14]"* As the M2M solutions, the IoT solutions are also based on a set of sensors and actuators that are connected to a network. However, unlike the M2M solutions, the IoT systems are also connected to the Internet. In this extended Internet, the real world objects can also communicate and interact as humans do in today's Internet, which enables countless application possibilities. These applications can, for example, create invaluable support and knowledge, which in turn can be used to improve enterprise processes. [13, p. 14-18]

Figure 2.2 represents the concept of the IoT. As can be seen, the IoT is composed of several smaller solutions, which resemble M2M solutions. Unlike the M2M solutions, they are connected to the cloud where the actual services and applications are running. When the applications can use and combine the data received from different sources, it is possible to create futuristic overall solutions, such as smarts cities. The term smart city is a concept where different infrastructures, such as heating, cooling, water, waste, energy, and transportation are integrated. This development brings the organizations of the city closer together, which makes services more efficient and eases the managing of the infrastructures. [13, p. 14-18]

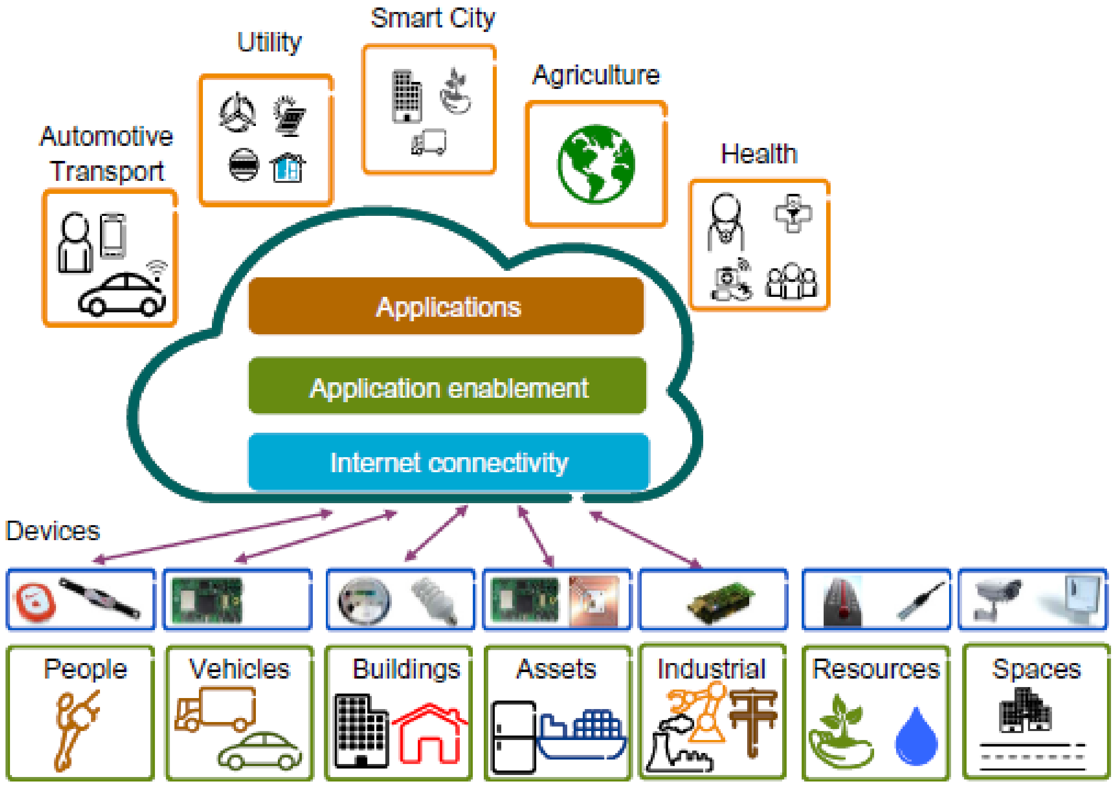

Figure 2.2. *Internet of Things [13, p. 15]*

In addition to the aforementioned smart city concept and the other IoT application segments presented in Figure 2.2, there are also many other application segments. For example, consumer electronics is one of the forerunners in the IoT world. Due to a countless number of different types of mobile devices and applications, people can use them for many purposes. For example, they can monitor their jogging route, heart rate, burned calories, steps, and many other parameters. All this data can be sent to the cloud where a service combines the data and provides clear reports, which can be shared on social media. [13, p. 14-18]

In the industrial world, the IoT has not yet became as popular as in consumer life. It is an interesting and widely discussed topic, but the actual IoT applications are still few and far between. However, there are many solutions of the IoT kind, which operate in more closed environments. These “Intranet of Things” solutions are becoming more popular and are pushing the industrial world towards the actual IoT. However, the competition between companies will probably keep these solutions at the “Intranet-level” for a long time to come. Some examples of these “Intranet-level” solutions are Sandvik’s Information management systems and the AutoMine product family. [13, p. 14-18; 25]

2.2 History

The term “Internet of Things” was used for the first time in 1999 by Kevin Ashton who is the co-founder and the executive director of the Auto-ID Center at Massachusetts Institute of Technology (MIT). At first, the term was tightly linked to the Radio Frequency Identification (RFID) technology but gradually it reach other technologies as well. [2]

In 2005, International Telecommunication Unit (ITU) produced a comprehensive report that reviewed the IoT from the technical, economical, and ethical points of view. One of the most significant offerings of the report was that it brought a new “anything” axis to complete the ubiquitous networking path, as shown in Figure 2.3. This axis increased the number of possibilities in the possible connectivity framework by adding new Machine-to-Person and Machine-to-Machine interactions in addition to the existing Person-to-Person and Person-to-Machine interactions. [4, p. 3-7]

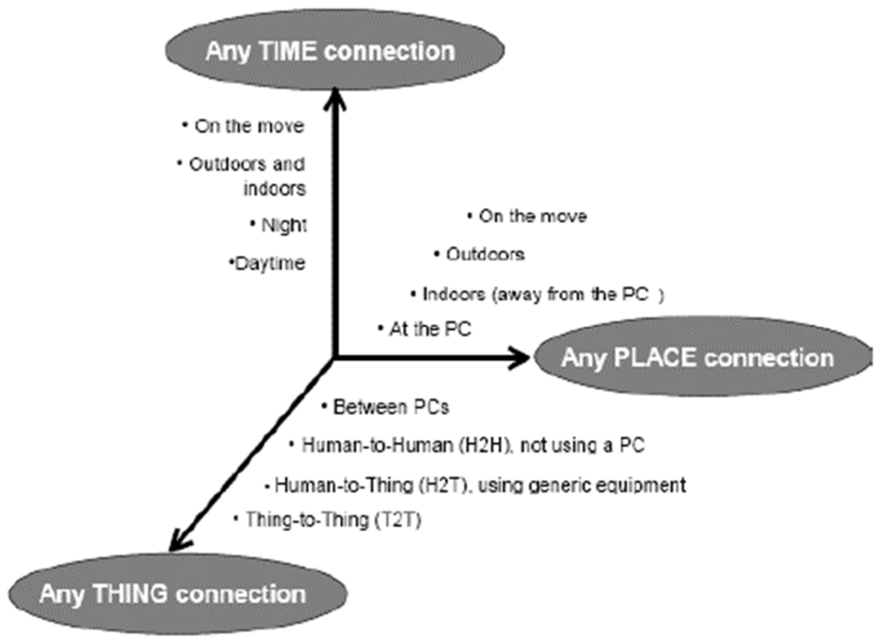

Figure 2.3. *ITU's vision of ubiquitous networking [4, p. 5]*

In 2008, the first international conference focusing on the IoT was held [8]. It was the first time when all the leading researchers and practitioners gathered to share research results and knowledge. After this, the international conference has been held in 2010, 2012, 2014, and 2015. Although the history of the IoT is longer, the growth of interest in the IoT was moderate until 2014 when the growth exploded. After that the Google searches volume has increased fivefold to the present day as can be seen from Figure 2.4.

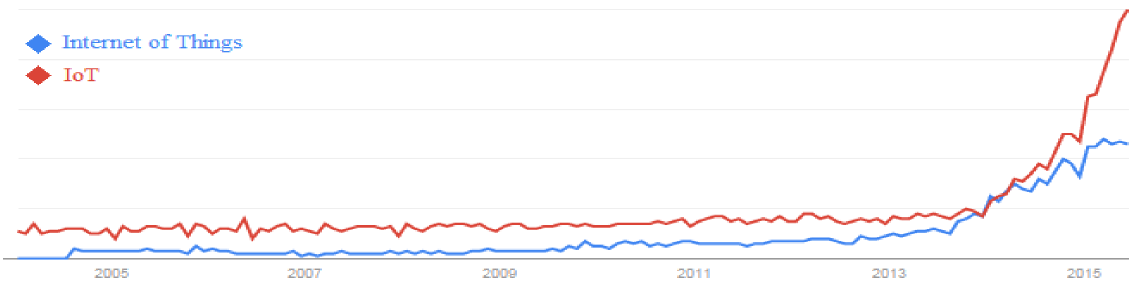

Figure 2.4. *Google search trends for "Internet of Things" and "IoT" [9]*

As the interest in the IoT, the number of actual IoT solutions has also grown radically during the past few years. Figure 2.5 shows the number of connected devices according to MIT business report (2014). As can be seen, the number of connected “things” increased from around 1 billion devices to around 3 billion devices between 2010 and 2014. The trend is expected to continue in such a way that there will be around 12 billion connected “things” in 2020. The increased number of “things” is considerably high if it is compared with the increased number of tablets, PCs, laptops, and mobile phones, which is expected to grow from around 11 million to around 16 million between 2010 and 2020. [21]

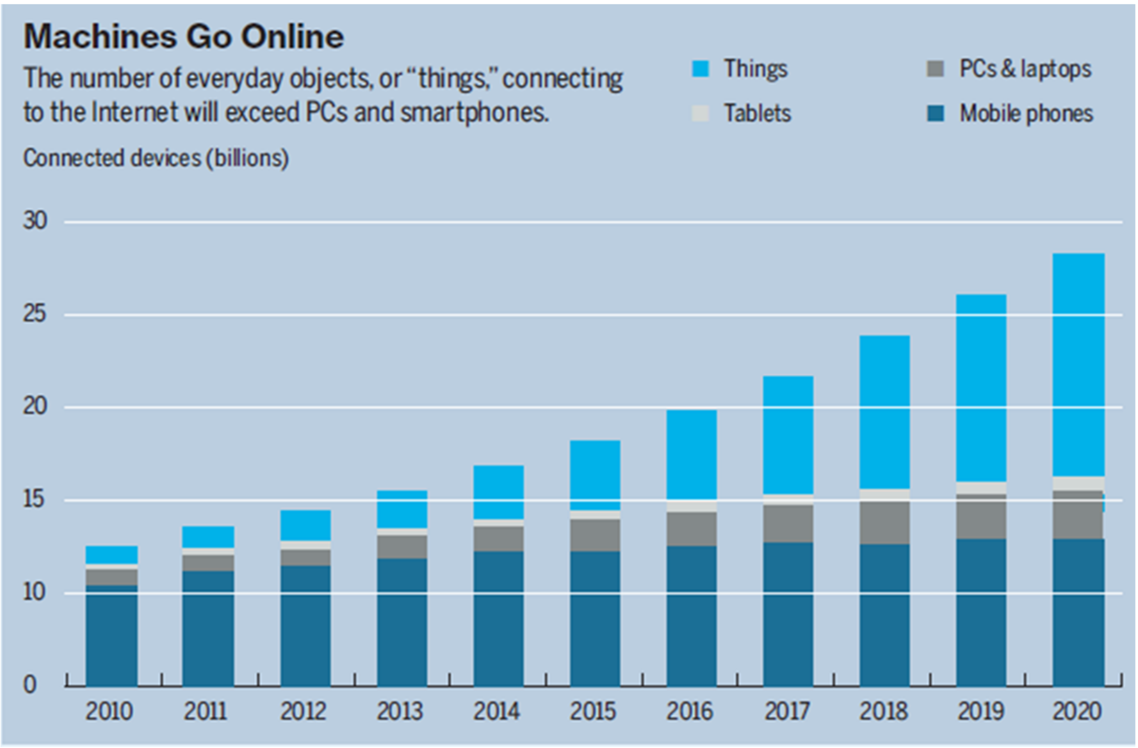

Figure 2.5. *The number of connected devices (2014) [21, p. 2]*

2.3 Drivers

Reasons given for the IoT trend vary depending on whom it is asked. According to Höller et al. the trend is caused by global megatrends that produce the required and enabling capabilities for the IoT business. These capabilities again create the IoT implications as represented in Figure 2.6. [13, p. 18-32]

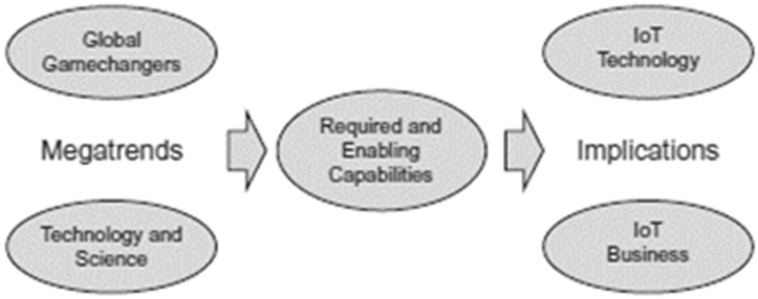

Figure 2.6. *Megatrends, Capabilities and Implications [13, p.19]*

The megatrends have been divided into the global game changers and breakthroughs of technology and science. The global game changers are global challenges or changes in society or economics that have a major impact on the future of people. These challenges and changes include, among others, the climate change, an increased interest in security and safety, urbanization, and many other similar phenomena. The other megatrend, technology and science, comprises the current technology and science trends, which create the enabling capabilities. These trends include, among others, trends in the field of material science, complex and advanced machinery, and Information and Communication Technology (ICT). [13, p. 18-32]

The trends in the ICT are core elements of the IoT and have created significant enabling IoT capabilities. The development of semiconductor components has enabled cheaper, smaller, and more capable processors, sensors, and actuators, which have facilitated the integration of the real world and the ICT. In addition to that, the great uptakes of IP protocol and wireless transfer technologies have enabled an instant and continuous connection to the Internet. [13, p. 18-32]

As already mentioned, the global game changers and the trends of technology and science create the required and enabling capabilities. The required capabilities are functions and features that are required or wanted to meet the challenges created by the game changers. For instance, an increased interest in safety creates a need for autonomous operations that reduce the need for human resources in hazardous environments. The enabling capabilities are technologies that enable the implementation of the required capabilities. For example, intelligent software solutions enable autonomous operations, and advanced sensing and actuation technologies enable remote control and monitoring applications. [13, p. 18-32]

The implications are the actual drivers of the IoT, which describe how megatrends and capabilities have affected the technology evolution and business attitudes. For example, a technological approach for systems has been moved from vertically to horizontal oriented, which means that companies want more open solutions instead of isolated solutions. This is a result of a trend where different organizations want to co-operate, share information and integrate their infrastructures. The horizontal system approach has also affected the business services by changing Business-to-Business type of service towards Business-to-Business-to-Customer type of service. [13, p. 18-32]

2.4 COSMOS

Keskustekniikka's COSMOS business unit has been a pioneer among the Industrial Internet of Things (IIoT) solutions, bringing the COSMOS product family to the market. The COSMOS products enable the IIoT and provide overall data collection solutions from sensors to a cloud. The system can be built to use existing logic controllers and sensors, but Keskustekniikka also provide their own devices. The collected data is stored in the COSMOS cloud service from where it is accessible from anywhere. [11; 18]

The main area of expertise of COSMOS is the marine industry where they, for example, have delivered data collection systems that help to optimize maintenance scheduling and efficiency. They provide the COSMOS products also for other industries where the systems might vary between a small scale single equipment system and a large scale facility system. [11; 18]

The current COSMOS products and services are presented in an example use case in Figure 2.7. COSMOS Instruments segment includes all sensors and detectors, which are parameterized according to needs and connected to upper-level systems. The COSMOS Connect service takes care of the security of the connection and is based on a Programmable Logic Controller (PLC) or a Distributed Control System (DCS) circuit. All the hardware components, such as an Ethernet switch and other automation components are located in the COSMOS Cabinets cabinet. The cabinets are delivered as "Turnkey"- solutions, which means that they are compatible for all customers and are immediately ready to use. COSMOS Collector takes care of data collection and recording. It also provides a local database and a wide range of different filters and calculations, which enable the monitoring of interesting technical information. From the local database, the data is forwarded to COSMOS Infosystem, which is a display and reporting tool that visualizes the collected data and takes care of the warnings and alarms of the system. The collected data can also be used for preventive maintenance that is the aim of the COSMOS Predict software. The software monitors the condition of an interesting device, which eases flexible and timely preventive maintenance. COSMOS Eyes in turn provides an image and sound system that enables intelligent surveillance or machine vision solutions. All the data is sent to COSMOS Cloud, which provides a common visualization, analysis, and reporting platform. The cloud service is based on the Microsoft Azure platform. [11; 12]

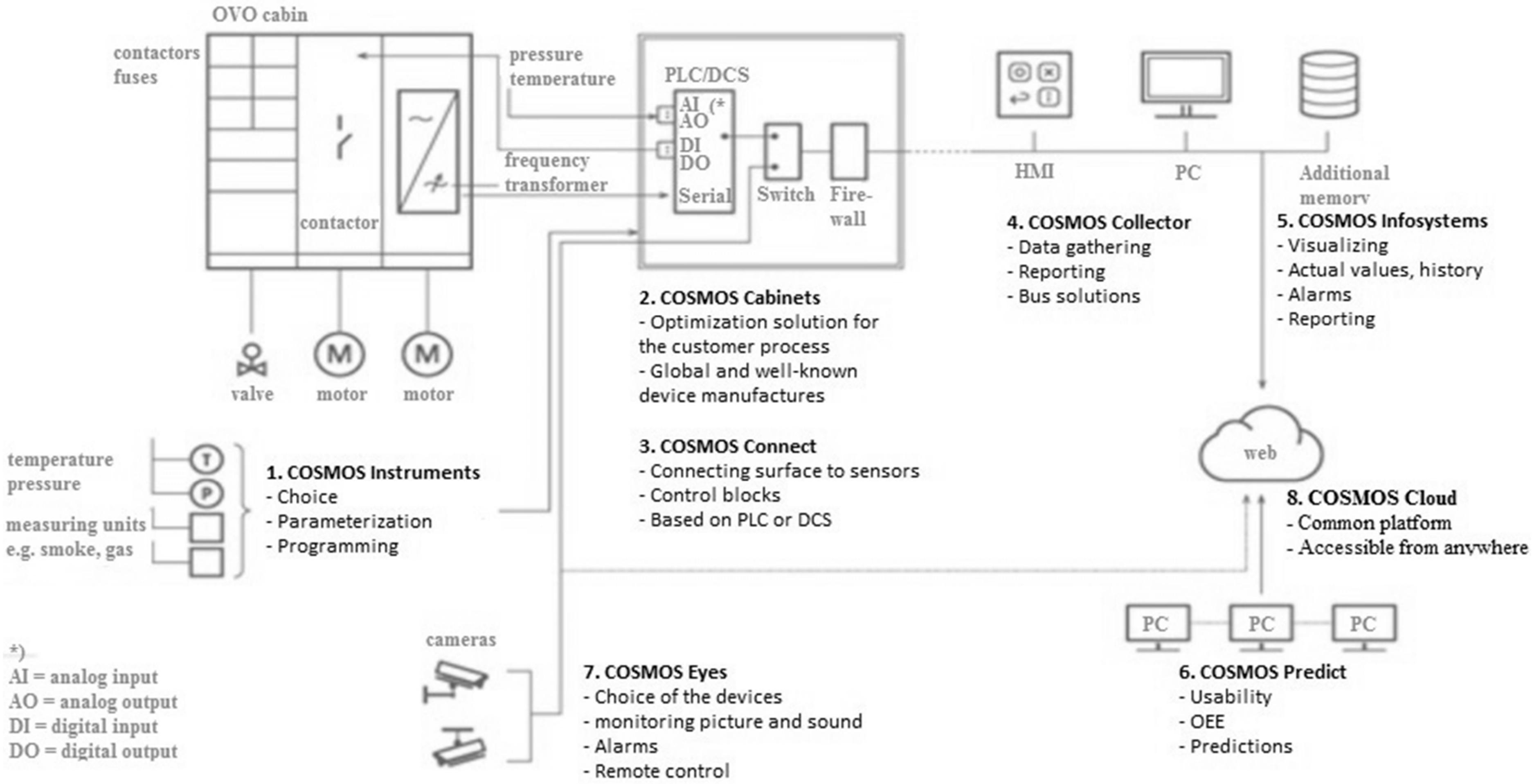

Figure 2.7. *COSMOS product and service family. Adapted from [11, p. 3]*

The aim of this thesis is to develop an intelligent data collection instrument that collects, utilizes, and stores interesting data and supports common data transmission technologies. The instrument is placed under the COSMOS Instruments segment, but it has many similarities with the COSMOS collector X10, which is introduced below.

The COSMOS Collector X10, in Figure 2.8, is a basic level data collection system that is based on an embedded system platform. It is built on the BeagleBone Black single-board computer and is capable of measuring around 500 measurement points. The X10 collects data from the source equipment with Modbus/TCP protocol via Ethernet connection. The collected data is analyzed with different calculations and stored in the database from where it can be forwarded to a cloud or other systems in XML or CSV format. As can be seen later, the features of the X10 are quite similar with the needs of Sandvik. For that reason, the X10 can be used as a reference for the instrument being developed in this thesis. [20]

Figure 2.8. *The COSMOS collector X10 [19, p. 1]*

3. IOT ARCHITECTURAL REFERENCE MODEL

The problem with the existing IoT solutions is that their architectures vary considerably. The solutions are designed and architected for specific applications in mind, and for that reason, they contain different technologies and protocols. The diversity of technologies and protocols causes the solutions not being able to communicate and operate together. This again conflicts with the vision of the IoT where all subsystems should be able to co-operate. Several different projects have tried to solve this issue by creating a blueprint and models to aid the designing of concrete systems. For example, European Framework Programme 7 (FP7) projects SENSEI (2013) and Internet of Things – Architecture (IoT-A, 2013), ETSI Technical Committee M2M (2013), and Industrial Internet Consortium (IIC, 2015) are projects of this kind [13, p. 63-64; 14]. This thesis uses the IoT-A project as its main source and introduces the IoT Architectural Reference Model (ARM). Its purpose is to establish a common understanding about the IoT and provide a common foundation for building interoperable IoT system architectures. [15, p. 21-25]

Figure 3.1 shows the structure of the IoT-A ARM and the process for generating a concrete ARM derived architecture. As can be seen, the IoT-A ARM consists of three parts: the IoT Reference Model, the IoT Reference Architecture, and the Guidelines. The IoT Reference Model and the IoT Reference Architecture give needed models, views, and perspectives for designing a concrete architecture and the guidelines guide how to derive and use those models, views, and perspectives. The design process uses the ARM as the best practices guide and together with the engineering strategies they guide the design process towards a concrete architecture that meets all the input requirements and use cases. [15, p. 35-45]

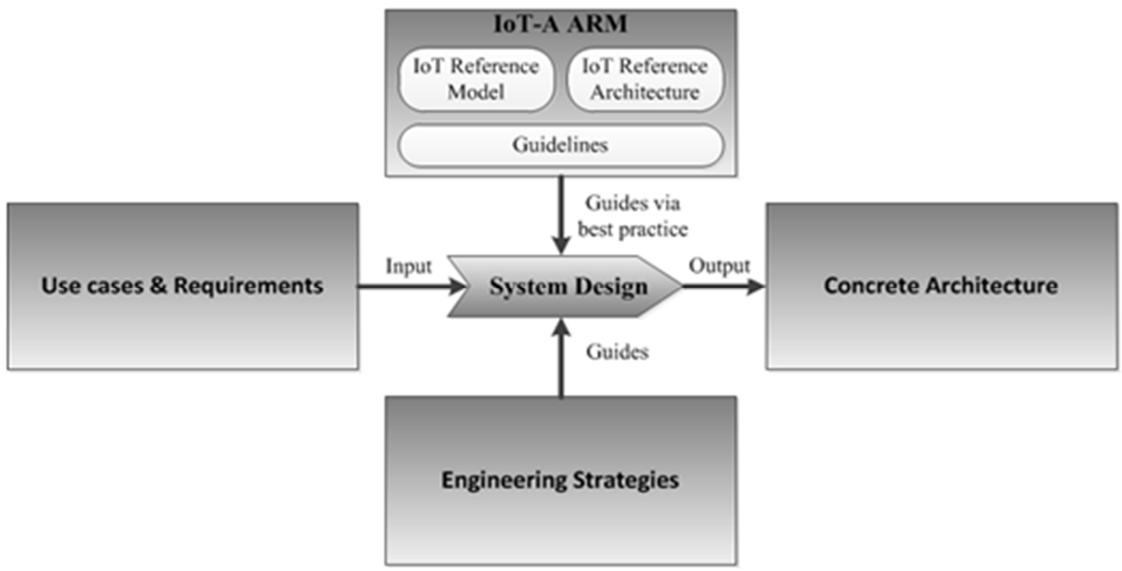

Figure 3.1. *Process for deriving a concrete architecture. Adapted from [16, p. 7, 15]*

3.1 IoT Reference Model

The IoT Reference Model defines the highest level of abstraction for an IoT system and provides all concepts and definitions that are needed for building an IoT architecture. As shown in Figure 3.2, the IoT Reference Model is composed of different sub-models of which the IoT Domain Model is the base for the other models. The IoT Domain Model describes the main idea of the system and introduces all concepts, such as devices, services, and relations between them. After the IoT Domain Model, there is the IoT Information Model that describes the information flow but not how it should be represented. The IoT Functional Model has been developed on the basis of the IoT Domain Model and the IoT Information Model. It is composed of Functionality Groups (FGs) of different types, which identify their type of functionalities and describe, for instance, the services and devices of a system. The last two models are the IoT Communication Model and the IoT Trust, Security & Privacy Model, which are tightly related to the FGs. The IoT Communication Model introduces the concept of communication, and the IoT Trust, Security & Privacy Model introduces trust, security, and privacy related functionalities and their interdependencies and interactions. [15, p. 46-48]

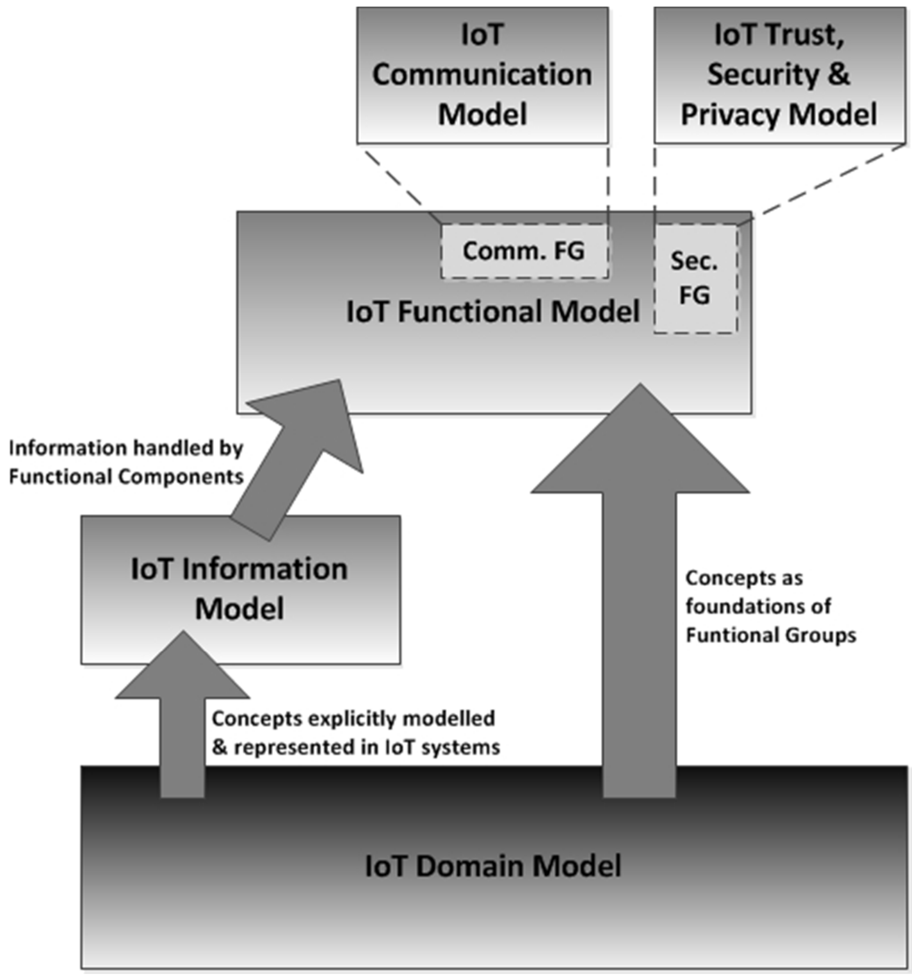

Figure 3.2. *IoT Reference Model. Adapted from [15, p. 47]*

3.1.1 IoT Domain Model

The IoT Domain Model is a base for all the other models and represents the core idea of the system so that it would be commonly understood. A common understanding is important because it eases architecting and enables various parties to argue about the solutions. To generate a common understanding, the IoT Domain Model introduces all main concepts of the system and defines the relations between them. The concepts are identified and named so that they can be recognized and their purposes can be understood. However, the level of details is kept so low that the model is relevant in the future. It means that the concepts are introduced at a theoretical level rather than at a technological level. For example, concepts can be identified as a tag rather than as an NFC tag, and as a user device rather than as an iPad. [15, p. 48-62]

The IoT Domain Model can be modeled using the Unified Modeling Language (UML), which is the modeling language standard that defines the most common diagram types. With these diagrams, a system design and architecture can be visualized and checked against the requirements before the actual implementation is started. This thesis does not cover the rules of the UML representation, but more information about the UML can be found on the UML website. [22]

The UML representation of the IoT Domain Model is shown in Figure 3.3. As can be seen, it represents the most important concepts of the system with blocks that are color-encoded according to the type of the concept. The blocks are contacted to each other with the arrows that illustrate relationships between the blocks. Because the relationships vary, the arrows are represented with different types of terminators and verbs that give further information about the relationships. The cardinalities, such as (0..*), illustrate how many target concepts the source concept can interact with and how many source concepts can interact with the target concept. For instance, (*0..* —acts— 0..*)* in Figure 3.3, illustrates that an Actuator may affect 0 or more Physical Entities, and a Physical Entity may be affected by 0 or more Actuators, whereas *(0..* —identifies— 0..1)* illustrates that a Tag may identify 0 or one Physical Entity, and a Physical Entity may be identified by 0 or more Tags. [15, p. 48-62]

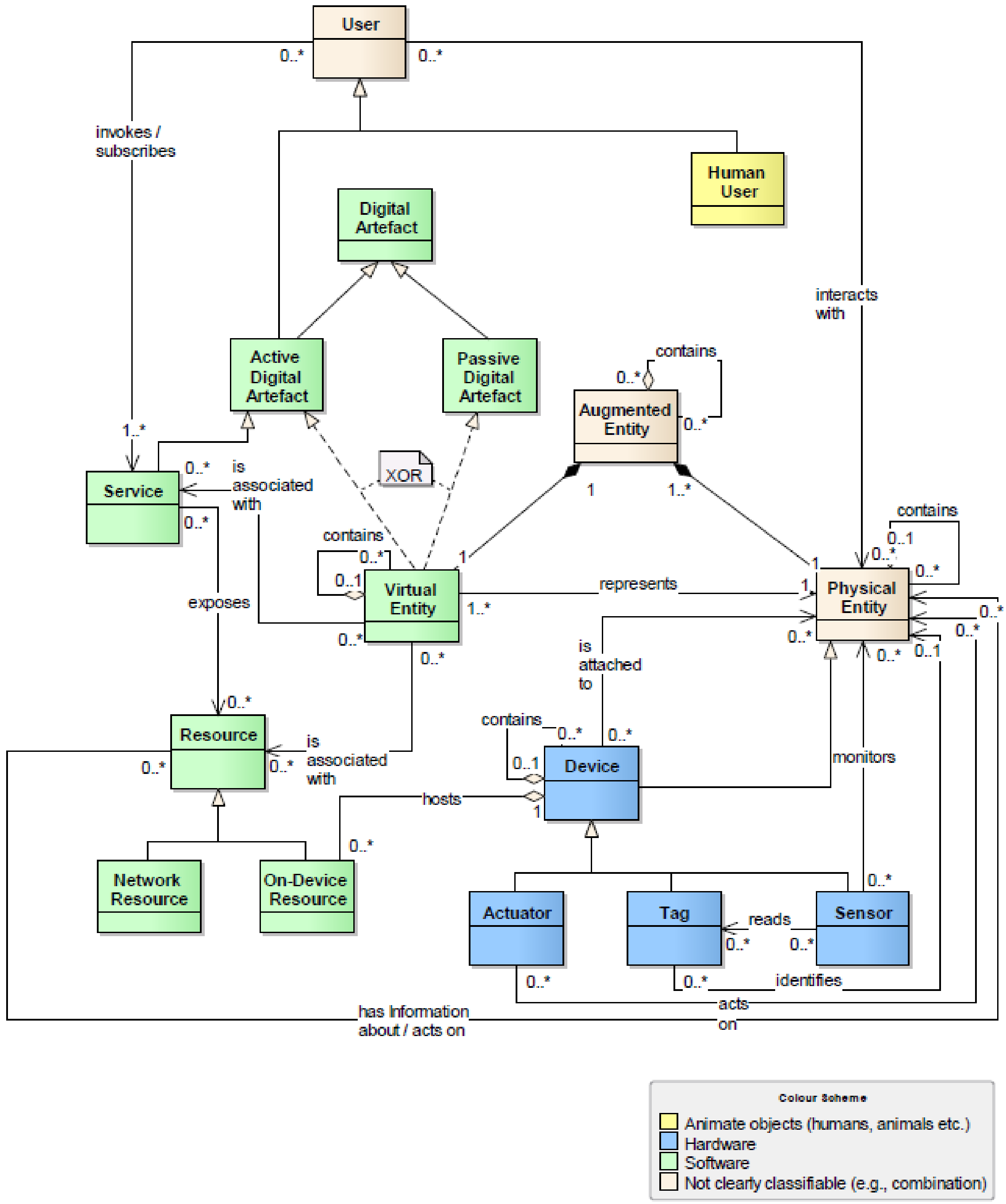

Figure 3.3. *UML representation of the IoT Domain Model [15, p. 58]*

In a concrete model, nonexistent concepts are left out from the model, and all the existing concepts are identified so that they can be recognized. It means that a concrete model does not necessarily include all the concepts presented above, but all the existing ones must be represented and named so that their purposes are understood. For example, a sensor can be named as Humidity Sensor or Temperature Sensor and a physical entity can be named as Warehouse. If a system is simple, all concepts can be drawn as separate blocks when the model does not need cardinalities. However, if the system contains, for example, multiple temperature sensors, the use of cardinalities simplifies the model. [15, p. 48-62]

3.1.2 IoT Information Model

The IoT Information Model describes the structure of the information related to the Virtual Entities of the system. It means that the IoT Information Model describes relations, services, attributes, and all other information that is represented and manipulated in the digital world. This model also gives a basis for representation, gathering, processing, storage, and retrieval of the IoT system's information and this way gives a basis for developing the rest of the IoT models. [15, p. 62-67]

Figure 3.4 represents the IoT Information Model in UML. As can be seen, the central concept is a Virtual Entity that represents a Physical Entity, such as a Warehouse, in the virtual world. A Virtual Entity or Entities may have an Attribute or Attributes that contain a Value and MetaData. For instance, a Warehouse may have an attribute, such as a Temperature that has a Value that in turn has a Timestamp. The Virtual Entity or Entities also contain a Service Description or Descriptions that describe the main details of each service. The Service Descriptions in turn may contain a Resource Description or Descriptions that describe the resource whose functionality is affected via the service. Furthermore, there may be a Device Description that gives information about the device on which the Resource is hosted. In some cases, there might also be an Association between a Virtual Entity and a Service Description. This Association either can read an Attribute value or set the value according to the service. [15, p. 62-67]

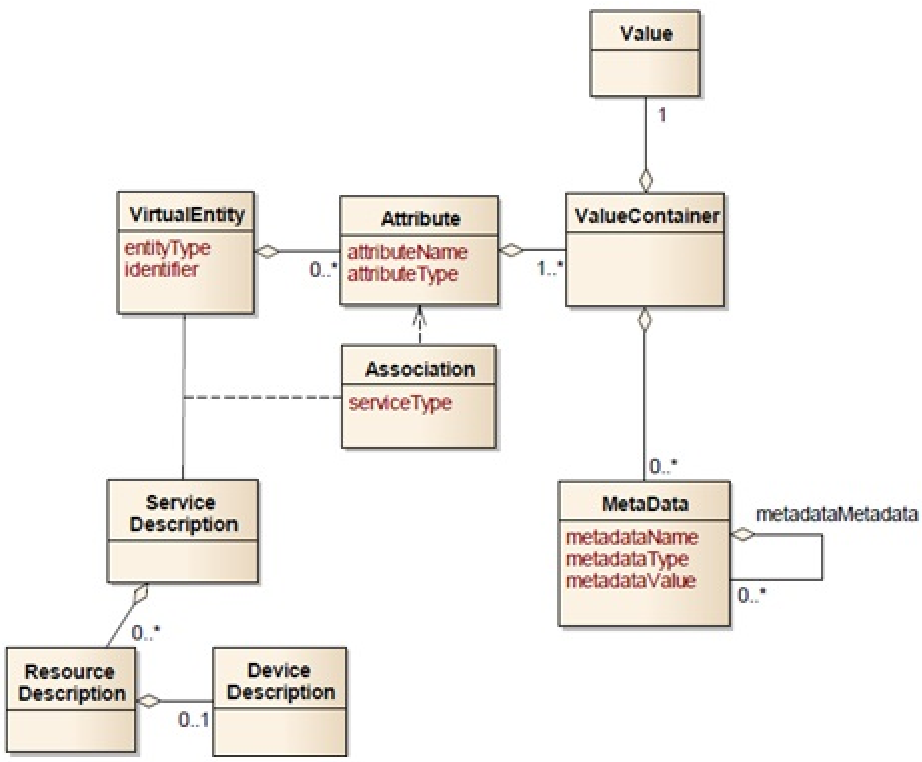

Figure 3.4. *UML representation of the IoT Information Model [15, p. 63]*

3.1.3 IoT Functional Model

The IoT Functional Model divides the main functionalities of the IoT system into nine Functionality groups (FGs). The model does not define any particular functionality component or technology, but it determines the FGs at the conceptual level and represents the interactions between them. The IoT Functional Model is used for the development of the IoT Functional View that is a more detailed description of the system functionalities. This view is part of the Reference Architecture and is introduced in Section 3.2.1. [15, p. 67-77]

Figure 3.5 represents the diagram of the IoT Functional Model. As can be seen, it is combined of seven longitudinal and two transversal FGs, which are connected with yellow arrows that depict the interactions between the FGs. The transversal groups contain functionalities that are required by each longitudinal group, which means that they interact with most of the longitudinal FGs. For that reason, the interactions between them are not depicted. The diagram alone is not enough to compose the IoT Functional Model, but the FGs need also more detailed descriptions. [15, p. 67-77]

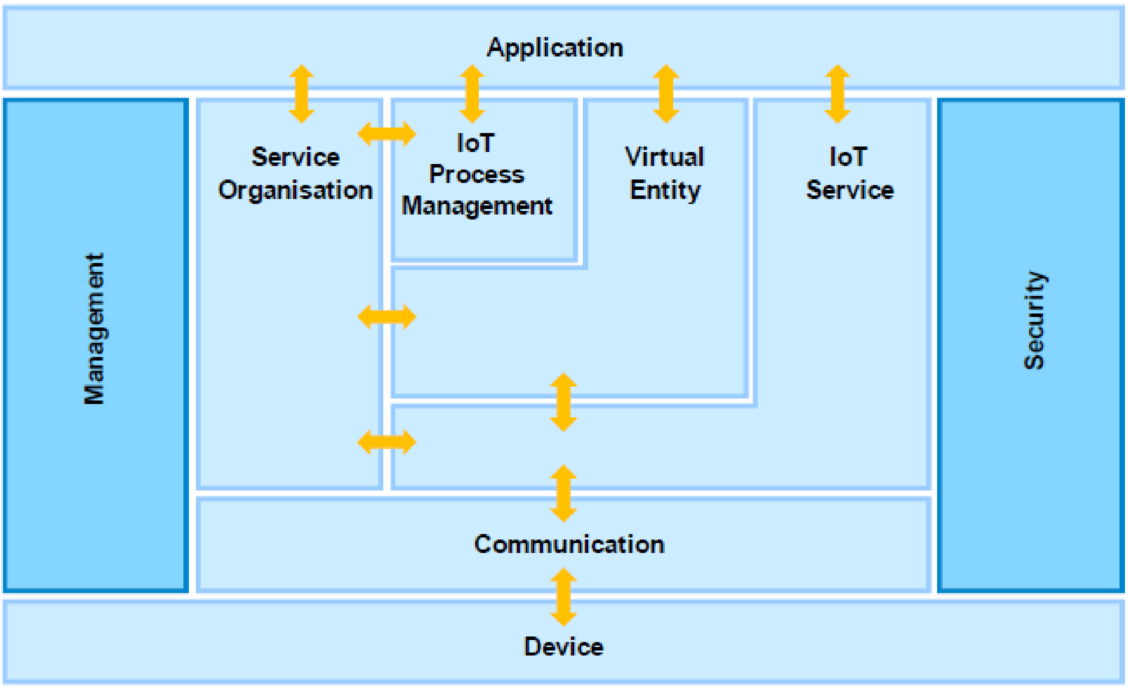

Figure 3.5. *The IoT Functionality Model diagram [15, p. 69]*

The descriptions of the Device and the Application FGs are the most generic ones. The Device FG describes the functionalities of the physical devices needed in the system. The functionalities include, among others, sensing, actuation, processing, data storage, and identification tasks. The Application FG in turn describes all the applications or sub-applications of the IoT systems and represents the logic they need. [15, p. 67-77]

The Communication FG introduces the communication technologies used for information transfer between digital world components and devices described in the Device FG. It provides a common interface for the IoT Service FG and considers, for example, following aspects: data representation, end-to-end path information, addressing issues, network management, and device features. [15, p. 67-77]

The IoT Service FG describes in more detail the services that are introduced in the IoT Domain Model. The group also introduces all support functions, such as discovery and look-up functions of the services. The IoT Service FG is tightly related to the Virtual Entity FG, which in turn describes the virtual entities in more detail. It, for example, describes functions and functionalities that interact with virtual entities or provide information about virtual entities. Furthermore, the Virtual Entity FG describes all functionalities needed for the association between the IoT services and the virtual entities. [15, p. 67-77]

The IoT Process Management FG combines the real world objects and processes with the IoT world. For example, it represents the functional concepts of the IoT system needed for the real world process management. The IoT Process Management FG relates tightly to the Service Organization FG that is a kind of communication hub of the IoT Process Management, Virtual Entity, and IoT Service FGs. The Service Organization FG composes and orchestrates different abstraction levels of services, and hence it provides a higher level service description. [15, p. 67-77]

As already mentioned, the functionalities of transversal FGs are linked to all other FGs. The Management FG describes functionalities needed for managing the IoT system. The goal of this management is to reduce costs, ease the control of unexpected issues and fault handling, and create flexibility. In addition to this, this group describes all the special rules of actuation if the system faces some interruptions, such as emergency stop or communication breaks. The Security FG describes the security functionalities needed to meet the safety requirements and to provide a safe and secure IoT system. For example, private parameter protection, secure communication, and authorized interaction are functionalities of this kind. [15, p. 67-77]

3.1.4 IoT Communication Model

The IoT Communication Model introduces concepts for communication. First, the model identifies all communication elements of the IoT system, after which it defines the communication paradigms of the interactions between the elements. The best paradigm for each interaction is selected according to the purpose of use. For example, unicast is usually used with traditional one-to-one communication, and multicast or anycast are used with many other applications, such as data collection and information dissemination applications. In addition to the purpose of use, the type of the communication element also affects the paradigm. When the interaction is between user-service or service-service, the interaction is usually implemented as a typical Internet interaction with standard Internet protocols. However, if the interaction is between service-resource or service-device, implementations might vary considerably. [15, p. 77-90]

This subchapter introduces two different ways to model the IoT communication. A simpler one is a channel model that does not include all possible details of the communication technologies but gives basic information about the interactions between the communication elements. Figure 3.6 represents the core idea of the channel model. As can be seen, the model depicts how data is received from a source and transmitted to a destination via a channel. The channel can be a simple network or a combination of several networks that are connected to each other with gateways. [15, p. 77-90]

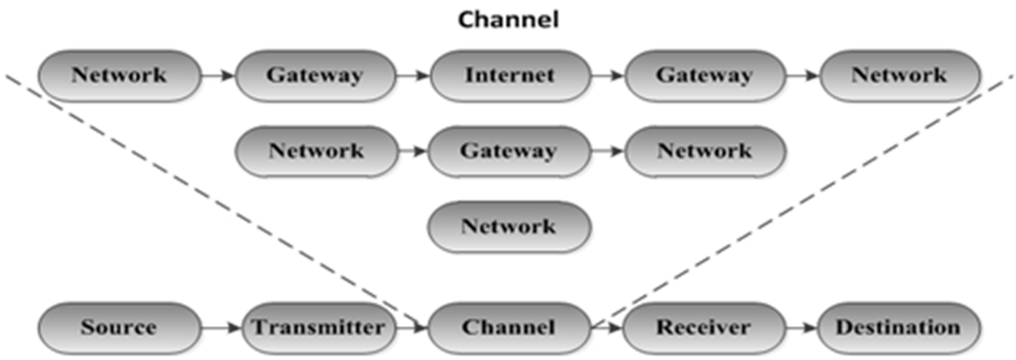

Figure 3.6. *The main idea of the channel model. Adapted from [15, p.89]*

The other model is a stack model that the IoT-A project prefers to the IoT Communication Model. This model is based on the ISO/OSI stack layer model (International Organization of Standardization / Open System Interconnection model) and describes different interoperability aspects of the IoT system. Figure 3.7 represents a graphical representation of this model together with the 7-layer OSI stack. As can be seen, they are quite similar, but the interoperability aspects combine some of the layers. This version of the IoT Communication Model provides a transversal approach where a single interoperability aspect can be used to describe interactions between different communication systems. [15, p. 77-90]

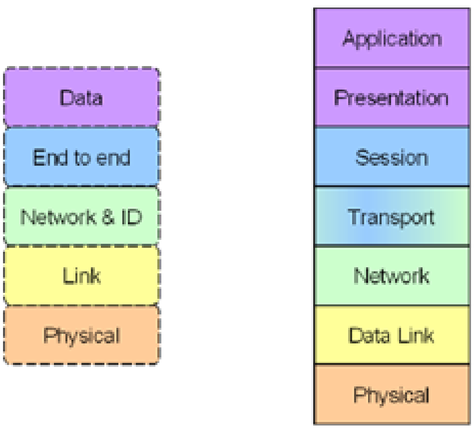

Figure 3.7. *The interoperability aspects compared to the ISO/OSI stack [15, p. 83]*

The Physical aspect focuses on the physical features of the communication technologies. This aspect restricts the communication technology options; however, it does not force the use of any specific technology. The Link aspect describes links between the physical and the network layer and provides standardized capabilities and interfaces. For that, it must describe several different functionalities that enable the coexistence of different data link layers. The Network & ID aspect concerns the network and the identification characteristics. It must provide a common communication paradigm for all communication technologies and make two technologies addressable from one another. The End-to-end aspect concerns reliability, transport issues, translation functionalities, and gateway supports of communication when connecting different network technologies. The Data aspect explains the data definitions and transfer techniques and models data exchange between the elements of the system. [15, p. 77-90]

Once the IoT Communication Model is modeled, the communication of the system is easy to represent as a set of OSI stacks. Figure 3.8 represents a graphical example of a gateway configuration where two end-point applications can communicate through two gateways. In this example, the first gateway bridges Ethernet and WiFi network. The other gateway bridges WiFi and ZigBee network and provides a translation functionality, which converts IP to 6LoWPAN, TCP to UDP and HTTP to CCoAP. [15, p. 77-90]

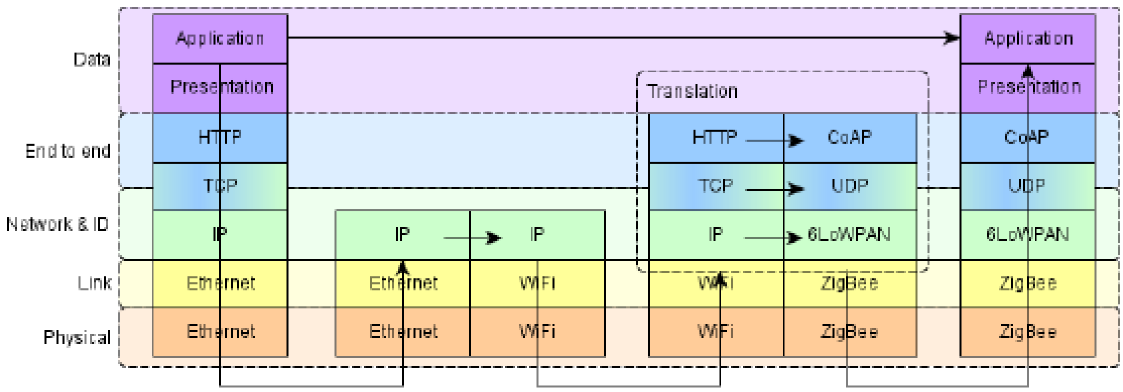

Figure 3.8. *Gateway configuration communication [15, p. 86]*

3.1.5 IoT Trust, Security & Privacy Model

After recent spying scandals it became obvious that these three properties are important in any IoT system, especially when a system contains sensitive data, such as personal information or trade secrets. These three models discuss potential security issues and describe the trust, security, and privacy aspects that guarantee these properties. [15, p. 77-90]

The aim of the IoT Trust Model is to ensure that all elements of the system behave as they should and as the other elements expect. For that, the IoT-A project describes six aspects that must be in the Trust Model:

1. Trust Model Domains aspect determines various trust domains, which contains a set of system elements that have similar trust properties. These domains ease maintaining trust relationships because defining the trust between each element separately is difficult or even impossible.
2. Trust Evaluation Mechanism aspect computes the trustworthiness degree of the system elements.
3. Behavior Policies aspect determines how the trustworthiness value affects the behavior of the system elements.
4. Trust Anchor aspect determines a system element that is trusted by default. It is usually used for evaluating the trustworthiness of third parties.
5. Federation of Trust aspect gives rules for handling relationships between system elements that use different trust models.
6. M2M Support aspect determines how autonomous machines can evaluate the trustworthiness of each other. [15, p. 90-92]

The IoT Security Model consists of three layers. The first layer is the Communication Security layer that divides the system elements into groups with similar communication specifications and describes the security functionalities of each group. The Application Security layer, first, detects and analyzes the possible risks of the applications, after which it guides the designing of the system in a way that damages and accidents would be avoided. The Service Security layer is similar to the Application Security layer. It detects and analyzes possible risks of services and guides the designing in order to prevent a realization of the risks. [15, p. 92-95]

The IoT Privacy Model describes the privacy mechanisms, such as access policies, encryption algorithms, and credentials. To ensure sufficient privacy, the model must contain the following components: Authentication, Authorization, Identity Management, and Trust & Reputation component. The Authentication component verifies the identities of the users. The Authorization component asserts and enforces the access rights of the users. The Identity management component ensures anonymity and unlinkability by providing multiple identities for one root identity. The Trust & Reputation component maintains the relationships between system elements. [15, p. 95-98]

3.2 IoT Reference Architecture

The IoT Reference Architecture plays an important role when designing a concrete IoT architecture. It aids design and gives a blueprint for any concrete IoT architecture. The IoT Reference Architecture has been created by studying existing IoT systems and their common architectural aspects. From these aspects, the best practices have been chosen, and the reference is constructed. Here, the IoT Reference Architecture is represented as a set of different architectural views and perspectives that focus on the system aspects that can be conceptually isolated. All views and perspectives follow the guidelines and standards of software engineering literature, which eases the work of architects. However, as some views in the literature are very use case dependent, they are excluded from the IoT Reference Architecture. [15, p. 102-105]

3.2.1 IoT Functional View

The IoT Functional View is derived from the IoT Functional Model and describes the Functional Groups further by introducing the Functional Components (FC). Because the Reference Architecture combines only the common aspects, it does not describe any use case- or application-dependent FCs. Also, the interactions between the FCs are not covered in the Functional View of the IoT Reference Architecture. [15, p. 105-126]

Figure 3.9 represents the IoT Functional View diagram of the IoT Reference Architecture. As can be seen, the Functional Groups of the IoT Functional Model are completed with the Functional Components that recur in numerous different IoT systems. The diagram gives a good overall view of the IoT Functional View, but the simple diagram provides only little added value; therefore, each FC requires a more detailed description that contains, for example, requirement mapping, system use cases, interaction diagrams, and interface definitions. These descriptions are very detailed and do not serve the aim of this thesis, which is why they are excluded from the thesis. However, more information about these descriptions can be found in the IoT-A project paper. [15, p. 105-126]

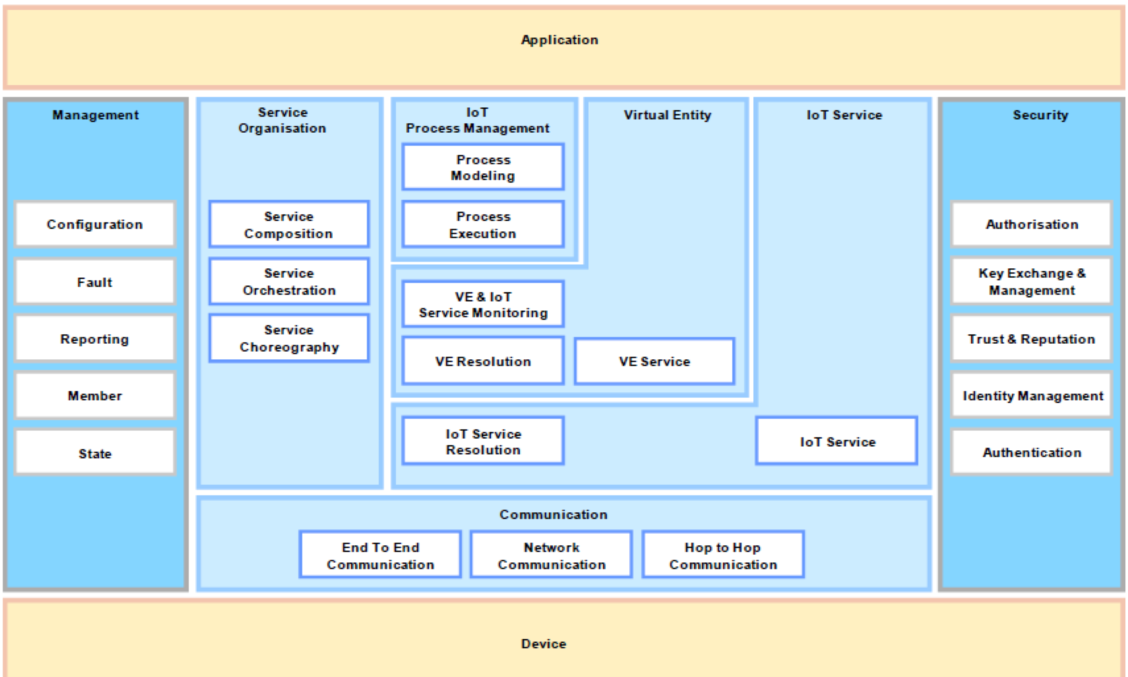

Figure 3.9. *IoT Functional View of the IoT Reference Architecture [15, p. 107]*

3.2.2 IoT Information View

The IoT Information View is based on the IoT Information Model and describes the static information structure and dynamic information flow in the IoT systems. The IoT Information View also defines the functional components that handle the relevant information, but it does not represent any concrete information representation alternatives. [15, p. 126-141]

The IoT Information view consists of two parts: the information description part and the information handling part. The first part describes the information handled in the IoT system and gives more details about the concepts introduced in the IoT Information Model. For this, the view describes, for example, an Entity Type model, Services on a syntactic and on a semantic level, and Associations between Virtual Entities and Services. [15, p. 126-141]

The second part describes the flow of information through the system and defines the life cycle of the information. Because the information flow depends largely on a use case, the IoT Information View of the IoT Reference Architecture introduces only four general information flow patterns and gives some examples how the Functional Components can use them. [15, p. 126-141]

Figure 3.10 represents the general information flow patterns. The Push-pattern is a one-way communication pattern, where a client pushes information to a server. The Request/Response-pattern in turn is a two-way communication pattern, where the client sends a request to the server. After the server receives the request, it sends a response back to the client. The Subscribe/Notify-pattern is also a two-way communication pattern, but in this pattern, the client first indicates their interest in a service by sending a subscription call to the server. After the subscription call, the server sends notifications to the client until the client unsubscribes the service. The Publish/Subscribe-pattern adds a broker component between the client and the server, which makes coupling between communication partners looser. [15, p. 126-141]

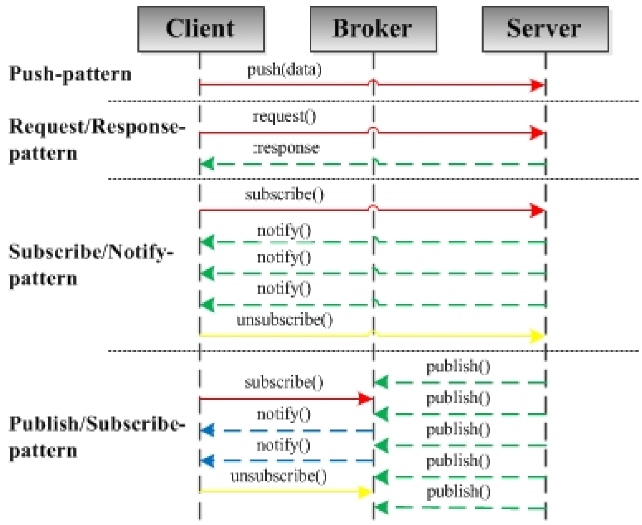

Figure 3.10. *General information flow concepts. Adapted from [15, p. 130-131]*

Figure 3.11 represents an example of how the Functional Components can handle information and use the information flow patterns introduced above. In this example, the Sensor Device pushes an updated sensor value to the IoT Service, which forwards the value to the Virtual Entity Service. The Virtual Entity Service and the User 1 uses Subscribe/Notify-pattern where the User 1 gets updates, for example, about the values of the Virtual Entity's Attribute. The User 2 in turn requests information from the IoT Service, which gives the corresponding response. [15, p. 126-141]

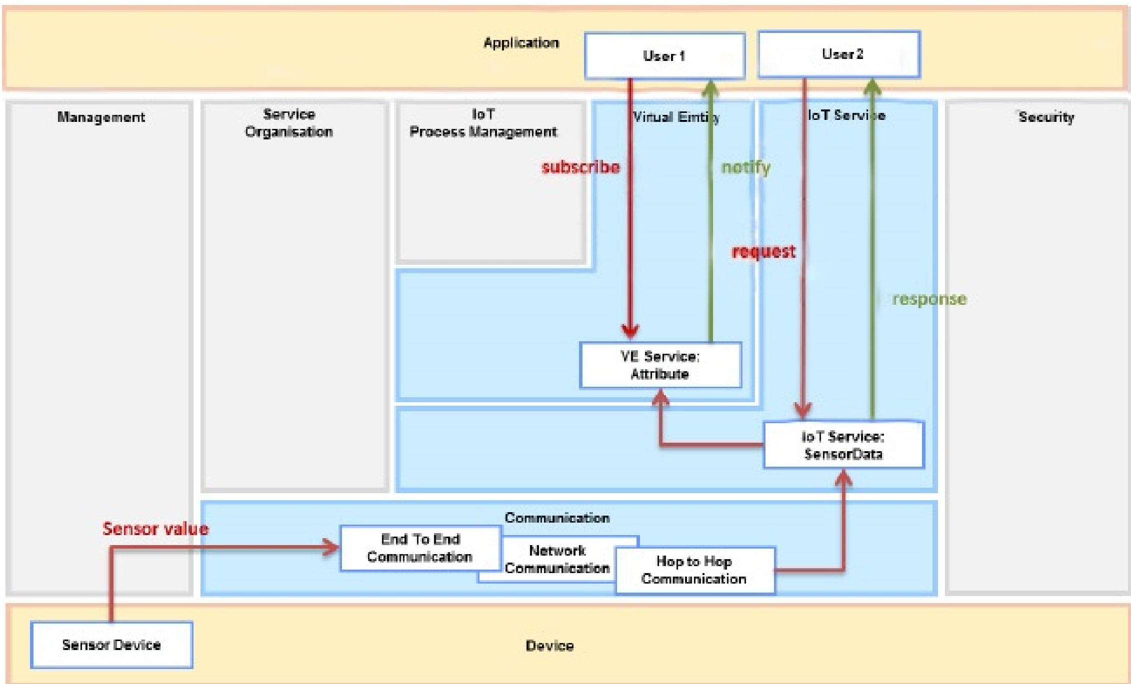

Figure 3.11. *Information flow example. Adapted from [15, p. 133-135]*

3.2.3 IoT Deployment & Operation View

The aim of the IoT Deployment and Operation View is to guide the realization of an actual IoT system. For this, the view discusses, for example, how to select the best technologies in order to ensure that the system is able to communicate and operate in a comprehensive way. Because the IoT Deployment and Operation View are strongly related to an actual system, the IoT Reference Architecture does not represent any diagrams. However, it identifies technology categories that have the strongest impact on the IoT system realization and this way gives guidelines for designing an actual implementation. [15, p. 141-149]

According to the IoT Deployment and Operation View the implementation of an IoT system can be divided into the following steps:

1. At first, a designer must decide what kinds of devices are used in a system. All the devices must respect the functionalities defined in the Functionality Model and View. In addition to this, the designer must also decide complexity levels and connectivity types of the devices.
2. After the device decision, the designer decides the communication technologies and communication protocols so that they respect the Functionality Model and the Communication Model.
3. The third stage is related to the system resources and services. In this phase, the designer decides if the software related to the resource is deployed in smart devices, on gateways, or in a cloud.
4. In the fourth step, the designer must decide where the data collected from the system is stored. In practice, there are three alternatives: It can be stored on a local device, on a web database, or on a local device with a web cache.
5. Finally, the designer's last choice is to decide the deployment of a core engine that handles the resolution of services and entities. It can be deployed either to the servers belonging to the system or then it can be provided by a third party. [15, p. 141-149]

3.2.4 Perspectives

The perspectives are set of activities, tactics, and guidelines, which relate to non-functional or quality properties that are common to more than one architectural view. Their purpose is to ensure that the particular properties are fulfilled in a real system. [15, p. 149-156]

The IoT Reference Architecture identifies following four perspectives that are the most essential for the IoT systems:

1. Evolution and Interoperability;
2. Availability and Resiliency;
3. Trust, Security, and Privacy; and
4. Performance and Scalability. [15, p. 149-156]

For each perspective, the IoT Reference Architecture describes the desired quality and applicability properties and gives the requirements that the perspective addresses. In addition, the perspective contains information about the possible activities and tactics that are suggested in order to achieve the desired qualities. [15 p. 149-156]

4. COSMOS INSTRUMENT FOR SANDVIK MOBILE EQUIPMENT

This chapter discusses the development process of the sensor-based COSMOS data collection system and describes how it can be integrated into Sandvik mining equipment. The main focus is on the data collection instrument (hereinafter referred to as COSMOS FX), but a server/cloud side of the system is also introduced shortly in order to clarify the operating principle of the complete system.

The first part of the chapter describes Sandvik's needs and requirements and represents the basic system models. The second and third part design the product concept for the instrument. The concept introduces higher-level hardware and software decisions, but do not select any exact hardware or software components. These higher-level decisions present how the instrument can be implemented according to Sandvik's requirements and provide a guideline for prototyping. The last part of the chapter introduces the server/cloud side of the system.

4.1 Background and Overview

As explained in Section 2.4, Keskustekniikka's COSMOS product family provides overall data collection solutions. The solutions vary, but especially their data collector COSMOS X10 has similar features that match Sandvik's needs. Sandvik's more specific needs and requirements are represented later in this section, but Sandvik is looking for a small, cost-effective, and intelligent data collector that works in all of their machines types. Because the features of X10 are similar to the wishes of Sandvik, it was assumed that by using X10 as a reference, an intelligent data collection instrument suitable for Sandvik's machines could be created.

Because Sandvik and Keskustekniikka have already worked in co-operation and thus know each other, it was sensible to co-operate also in this project. This co-operation profits both parties: Keskustekniikka can test their systems in Sandvik's machines and expand their COSMOS product family, and Sandvik receives information about sensor-based data collection solution options. In the best case scenario, Sandvik will have a new and functional data collection system as a result of the co-operation.

4.1.1 Identification of Need

Different data collection systems have become more common, providing interesting information about targeted variables. Also, Sandvik is constantly looking for new and alternative ways to gather usage data from their mining machines. They already have AutoMine® Process Management and AutoMine® Monitoring systems, which are wider data collection systems which have additional features, such as condition and health monitoring and task management. In addition to their existing systems, Sandvik wants a simpler and cheaper option that could be used in their old and new machines, regardless of the type of the machine. With this simpler system, they could accelerate the spread of their data collection and monitoring systems and this way gather more interesting information about the usage of their machines. A few examples of Sandvik's mining machines are presented in Figure 4.1. [27]

Figure 4.1. *Machine type examples. Adapted from [25]*

One option for this simpler system is an intelligent sensor-based data collection instrument that is the key point of this thesis. Several manufacturers already have different types of sensor-based solutions that are mainly used for remote condition and utilization monitoring of a static equipment. Because Sandvik's machines are mobile machines that are used in rugged operating environments, none of the existing solutions are directly suitable for these machines. The existing solutions, however, give a reason to believe that sensor-based solutions can be also used in this use case, which COSMOS FX aims to achieve. [27]

The use of this new system depends on the final product, but the possible business models can be divided into two models. In the first model, Sandvik would sell the system directly to end-customers, who could use the system for utilization monitoring and production and maintenance planning. This model presumes that the system would contain a web-based user interface that graphically represents the usage data of the monitored machines. [27]

In the second model, Sandvik would use the data for their own purposes and then the system would not need an independent user interface. The data could be used for example:

1. To sell maintenance to customers who needs it.
2. To ensure the effectiveness of the warranty programs.
3. To monitor the utilization percent of Sandvik machines.
4. To sell analyzed data to the end-customers. [27]

4.1.2 Sandvik's Requirements

Sandvik's requirements for the system are presented in Table 4.1. Each requirement contains an ID number, a short description, a classification according to necessity, and remarks. The mandatory requirements include those features and requirements that must be built into the system, and the optional requirements include those features and requirements that should or can be built into the system. In addition to these technical requirements listed in Table 4.1, the target cost is also one of the significant factors for this system. [27]

Table 4.1. *Sandvik's requirements [27]*

ID	*Requirement Description*	*Mandatory / Optional*	*Remarks*
R1	The instrument's power supply shall be 9-36 VDC.	M	Alternative power supply options can be considered
R2	The instrument shall be overvoltage protected.	M	
R3	The instrument shall be at least IP67 sealed (including connectors).	M	M12 connectors wanted.
R4	The instrument shall be able to install in 15-30 minutes by an inexperienced person.	M	Including configuration.
R5	System shall work in Sandvik machines of all types (loaders, drill rigs, crushers…)	M	This thesis focuses on a couple type of drill rigs.
R6	The instrument shall be able to storage 6 months data.	M	
R7	The instrument shall contain a real time clock.	M	
R8	The instrument shall transfer data automatically to a main database	M	
R9	Data transfer shall be able to do manually with USB, NFC, Bluetooth or with any another technology.	M	
R10	The configuration must be as simple as possible, and if possible, the instrument should not need any instrument/machine-specific configuration.	M	Teaching buttons can be used for operation mode configuration
R11	System shall record: idle hours / engine hours / operating hours	M	
R12	System should record / detect: power pack on hours / which boom is drilling	O	
R13	The instrument should contain LED lights that indicate the status of the instrument.	O	
R14	The instrument should have possibility for external sensors.	O	
R15	The system should be able to connect to the cloud in the future.	O	The COSMOS software already has this capability.
R16	The system should contain reporting system that generates clear graphical reports.	O	Beyond the scope of the thesis.

4.1.3 System Models

Creating an IoT ARM-derived architecture was based on a set of models and views as presented in the previous chapter. This section presents the most important models that represent the operating principles of the system. Introducing the models also help a reader to understand the system.

The Domain Model of the system is represented in Figure 4.2. As can be seen, its Physical Entity is a Mining Equipment, which is also represented as its Virtual Entity. The COSMOS FX instrument (Device) is attached to the Mining Equipment, and it contains a Sensor that measures a Measurement Parameter of the Mining Equipment (On-Device Resource). The instrument is hosting an Equipment Mode Service (Service) that concludes an operation mode of the Mining Equipment from the Measurement Parameter. An Application, which is also running on COSMOS FX, uses the Equipment Mode Service. It indicates the operation mode and sends the operation mode data to a Database (Network Resource). The Database is exposed for a Reporting System (Network Service) that generates reports and all other statistics needed. Both, the reporting system and the Database can run either on a local server or in a cloud.

In normal use, the system has two types of users: A First User, who installs and configures the instrument, and a regular User, who uses the reporting system in order to view reports and other statistics. There might be also a user who manually transfers the operation mode data to the Database, but this user is left out from this model.

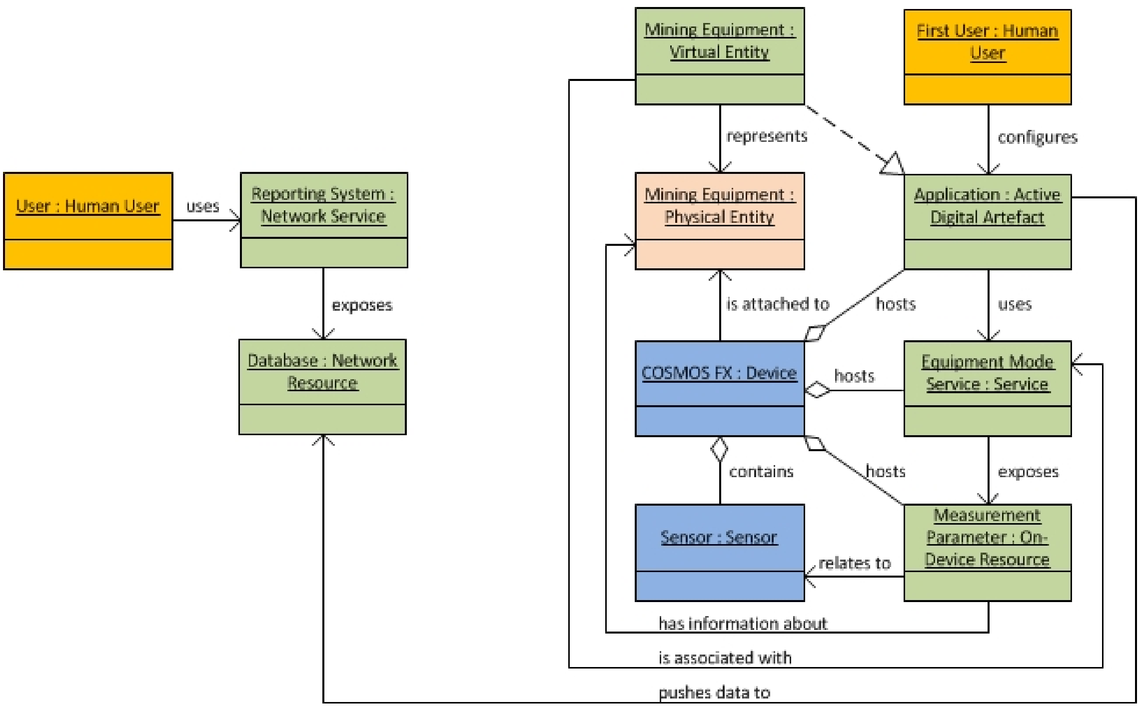

__Figure 4.2.__ Domain Model of the system

The IoT Information Model modeled for this use case is shown in Figure 4.3. It represents the structure of the information that constitutes the Virtual Entity and describes the service that acts on the Virtual Entity. The Virtual Entity is the Mining Equipment, which has an Operation Mode Attribute. The Operation Mode Attribute has a Value that indicates the current operation mode of the Mining Equipment and a Timestamp (Metadata) that indicates when the Value is detected. The Value of the Operation Mode is received from the Equipment Mode Service that accepts a Measurement Parameter (Resource) as an input and gives the current Value as an output. The Measurement Parameter is measured by the Sensor.

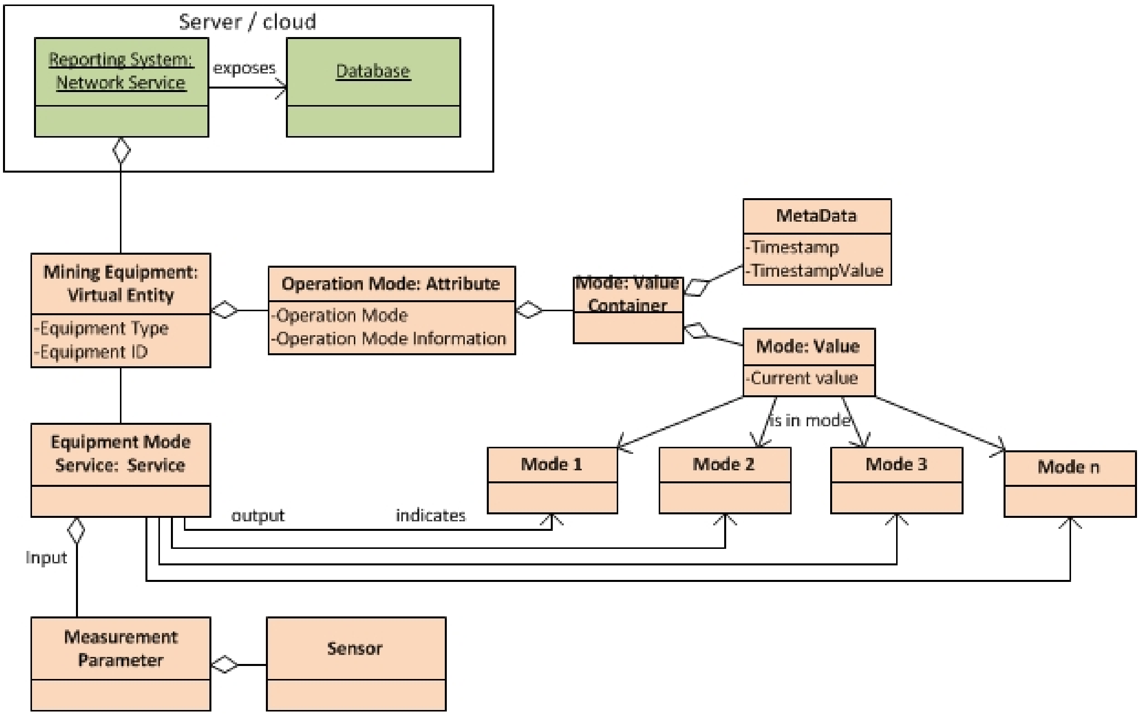

Figure 4.3. *Information Model of the system*

The Functional Components of the system are shown in the Functional View in Figure 4.4. The components that system uses during normal runtime are indicated in orange, and the components that are only needed when the system is built are indicated in yellow. The Hop to Hop Communication is indicated in orange-stripped. It demonstrates the possible hop to hop communication, which is one implementation option for manual data transmission. The precise description of the Functional Groups and the Functional Components would require dozens of the system use cases, interaction diagrams, and interface definition be presented, nut they are excluded from this thesis. However, some use cases and process models are presented in Section 4.3 to support the description of the software.

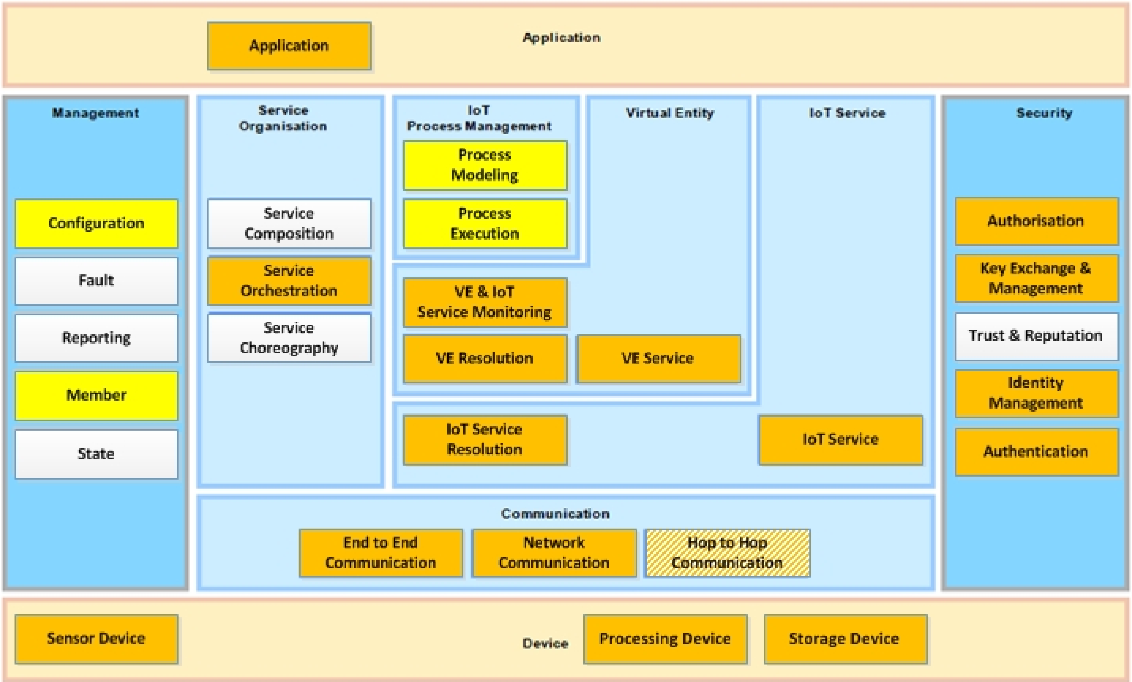

Figure 4.4. *Functional View of the system*

4.2 Hardware of COSMOS FX

This section represents the most significant hardware decisions that outline the plan to build the instrument. Before any other decisions, it must be decided whether to build a brand new instrument or to use an existing device. The use of COSMOS X10 with additional sensors would be the simplest option, but it is not the most cost-effective solution. Another option is to use some other manufacturer's device, such as Yepzon GPS locator. Although Yepzon's device is intended for locating services, it also contains an accelerometer that can be used for operation mode detection [29]. This option would increase dependence on other parties and would decrease the possibility of influencing the design of the instrument. For these reasons, a brand new instrument is considered the best option.

4.2.1 Platform

As the COSMOS collector X10, COSMOS FX is built on a Linux-based embedded system platform. Table 4.2 shows the possible platform options with their main features. When the features, such as a price, speed, memory, connectivity options, and a number of pins are compared, it becomes obvious that Teensy 3.1 is the best choice for this instrument. It is the cheapest option that is, however, more powerful and has more memory and pins than Arduino. Raspberry and BeagleBone, in turn, contain many unnecessary features that only raise the price.

Table 4.2. *Hardware platform options [1; 6; 23; 24; 26]*

Platform	***Arduino Uno***	***Teensy 3.1***	***Raspberry Pi 2 Model B***	***BeagleBone Black***
Micro-controller	ATmega328P	Cortex-M4	ARM Cortex-A7	ARM Cortex-A8
Clock Speed	16 MHz	72 MHz / 96 MHz (over-clocked)	900 MHz	1 GHz
Input Voltage	7-12 V	5 V tolerant	5 V	5V
Pins	14 digital I/O 6 analog inputs	34 digital I/O 21 analog inputs	40 GPIO pins	2x46 pin headers
Flash Memory	32 KB + optional microSD card adaptor	256 KB + optional microSD card adaptor	MicroSD	2GB eMMC + microSD
RAM	2 KB	64 KB	1 GB	512 MB
Connec-tivity	USB Serial SPI I2C	USB 3xSerial SPI 2xI2C CAN I2S audio	4xUSB 2.0 Serial SPI 2xI2C Ethernet HDMI 3.5mm audio LCD (DCI)	2xUSB 2.0 1+4xSerial SPI I2C CAN Ethernet HDMI McASP LCD
Software	Arduino Tool	Teensyduino	Windows 10, Debian GNU/Linux, Fedora, Arch Linux, RISC OS, and more	Debian, Android, Ubuntu, Cloud9 IDE on Node.js with BoneScript, and more
Price	24.95 $	19.95 $	39.95 $	54.95 $

4.2.2 Power Supply

For the power supply, Sandvik has two requirements. The requirement R1 requires the COSMOS FX instrument to operate on 9-36 VDC input voltage, and the requirement R2 requires overvoltage protection. These requirements are related to the traditional wired power supply solution, but a few other solution options, such as a battery and an energy harvester, are also considered as possible power supplies.

The idea of the energy harvester solution is represented in Figure 4.5. In this solution, a vibration harvester would convert the vibration energy of a machine to an electrical energy for the COSMOS FX instrument. For the energy conversion, the solution would need a piezoelectric vibration harvester that would be located outside the instrument because the harvester itself might cause resonance on the vibration sensor. The solution would also contain a battery and power management component, which optimizes the power for Teensy. This solution provides an interesting option, but is rejected due to complexity and expensiveness.

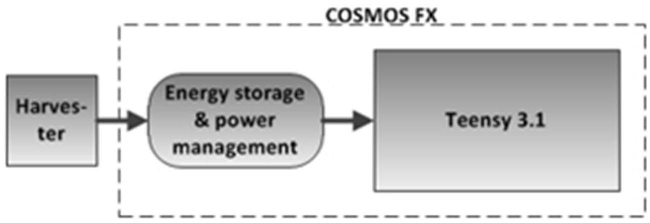

Figure 4.5. *Energy harvester solution*

The second option is to power the COSMOS FX instrument using a battery, which would be either integrated into the instrument or installed into an external battery pack. The idea of the integrated battery solution is represented in Figure 4.6. In this solution, the whole instrument would be changed during the manual data transfer, after which it would be recharged and installed on another machine. Changing the whole instrument is considered impractical, and for that reason, the integrated solution is rejected.

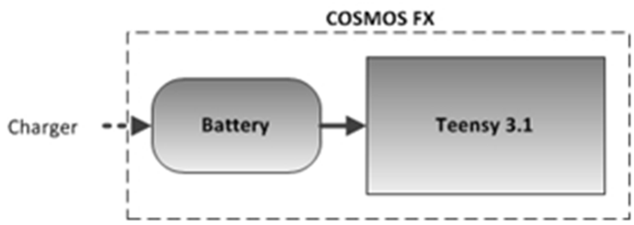

Figure 4.6. *Integrated battery solution*

The external battery solution is represented in Figure 4.7. In this solution, only the battery pack would be changed during the manual data transfer. Because the data transfer interval might be longer than the discharge time of the batteries, this solution is also rejected as the primary power supply. However, the battery pack can still be used as an accessory.

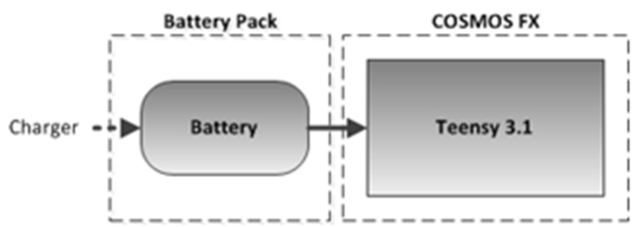

Figure 4.7. *External battery solution*

The wired power supply solution is the best choice for the instrument. The main idea of this solution is represented in Figure 4.8. As can be seen, the power for the instrument comes from a DC circuit of a machine, from where it is supplied through a voltage regulator to the Teensy board. According to the requirements R1 and R2, the voltage regulator is selected to be of a type that converts 9-36VDC input to 5 VDC output, and contains an integrated overvoltage protection circuit (OVP). The power connector is selected to be of type M12, as the requirement R3 requires.

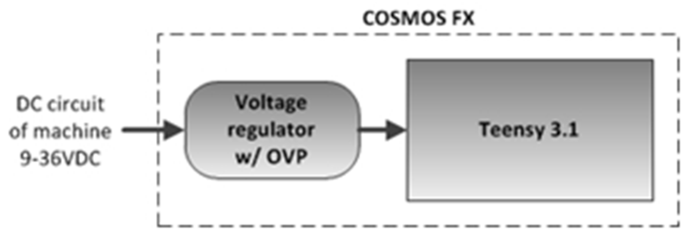

Figure 4.8. *Wired power supply solution*

4.2.3 Data Collection

Data collection is the main feature of the COSMOS FX instrument, and for that reason, the collection technique must be chosen carefully. Data collection from a computing bus is rejected due to a large variety of machine and computing bus types. Even if the wanted data could be collected directly from a bus, the system would need a more complicated configuration, which is against the requirement R10. In addition, if COSMOS FX was connected to a bus, the installation of the instrument would be more complicated, which in turn is against the requirement R4. Because of these reasons, the best option is to collect the data with internal sensors.

For the internal sensors, there are two possible options: a vibration sensor and a sound detector. Both of them convert a physical phenomenon into an electrical voltage signal that is proportional to a detected sound or vibration. By computing Fast Fourier Transform (FFT) of that signal it is got the component frequencies, which can be used for operation mode detection, as explained in Section 4.3.3. Because the sound detector might muddle the sounds of adjacent machines, the vibration sensor is the best option. Core idea of the FFT is explained in Appendix A.

The vibration sensor's ability to detect different operation modes was tested in preliminary workability tests and the results are presented in Appendix B. The results show that the tested vibration sensor can detect vibration clearly, and different frequency components can be distinguished. The tests also show that a whole frame of a machine vibrates at the same frequency, which means that the location of COSMOS FX does not affect the detected frequency components.

In addition to the internal vibration sensor that is connected to Teensy's I2C bus, the requirement R14 requests a possibility for external sensors. This requirement is met by adding a 5-pin M12 connector whose pins are connected to Teensy's input ports. In this way, it is possible to collect additional data, for example, from an additional sensor or a relay of a mining equipment.

4.2.4 Data Transmission

Due to the challenging operating environment, data transmission from COSMOS FX to the COSMOS server/cloud is not as simple as it would be in many other targets. Underground tunnels block radio signals effectively, and building a new comprehensive network only for this system is not a sensible option. Using the network of an end-customer, in turn, would not necessarily be possible or permissible. For these reasons, innovations for data transmission are needed.

The requirement R8 requires automatic data transmission from COSMOS FX to the COSMOS server/cloud, and the requirement R9 requires an alternative manual data transfer possibility. The automatic data transmission might be possible to implement with wearable solutions or with smart devices so that an operator of a machine would transfer data to the server/cloud without performing any actions. In this implementation plan, the automatic data transmission is, however, implemented with Wireless Local Area Network (WLAN), leaving the alternative options for further development.

Figure 4.9 shows the operating principle of the WLAN data transfer method. As required, the collected data is transferred automatically whenever COSMOS FX is within a WLAN coverage area. The existing COSMOS software already contains an automatic WLAN data transfer feature, and for that reason, the implementation of WLAN is easy and requires only a WLAN module that is attached to a serial port of Teensy. Although the implementation itself is easy, the problem with WLAN is that it needs a network configuration, which is against the requirement R10 that requires a simple and non-instrument-specific configuration. The implementation of the network configuration is covered in Section 4.3.2.

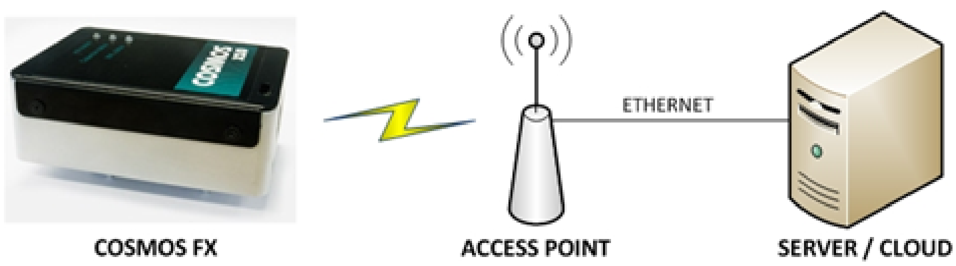

Figure 4.9. *Main idea of the WLAN method*

The operating principle of manual data transmission is presented in Figure 4.10. The main idea is that the collected data is transferred from COSMOS FX to the COSMOS server/cloud manually with a transfer device. First, an operator of a machine or an employee of Sandvik downloads the collected data to the transfer device from where it is uploaded to the COSMOS server/cloud.

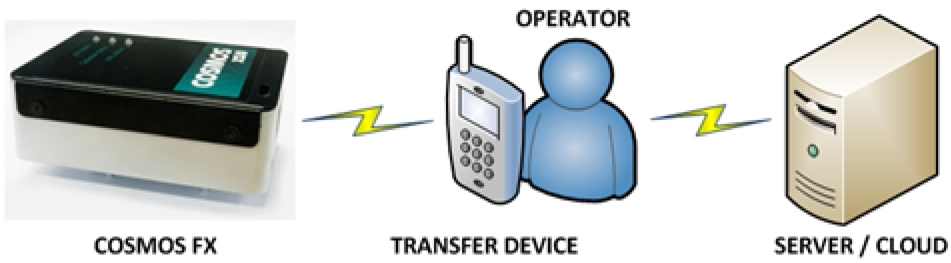

Figure 4.10. *Main idea of the manual data transmission method*

The technology options for manual data transmission are Bluetooth, NFC, and USB. The main features of each technology are presented in Table 4.3.

Table 4.3. *Technology options for manual data transmission*

Technology	*Range*	*Price*	*Pros.*	*Cons.*
Bluetooth	10m [3]	High [26]	Good range	Needs pairing and Bluetooth device for data transfer
NFC	0-5cm [10]	High [26]	Easy to use	Short range. Needs an NFC device for data transfer
USB	-	Low [26]	Cheap. Large data capacity	The COSMOS FX instrument needs a USB port.

Bluetooth and NFC would provide a wireless data transfer possibility when COSMOS FX would not need any external connectors for data transfer. Bluetooth's range would also enable freer placement for the instrument because the data could be transferred a few meters away from the instrument. The transfer device with Bluetooth and NFC could be a smartphone or some other Bluetooth or NFC device, such as a smartwatch or an NFC card. However, expensiveness and the fact that use of mobile phones is restricted at some mine sites are the reasons why USB is the most suitable option in this case. It is the cheapest option that still performs its task and meets Sandvik's requirements. Also, the wide availability of USB flash drives affects the decision. According to the requirement R3, the USB port type must be M12.

4.2.5 Other Components

The other components of the COSMOS FX are the components that were not covered in the previous chapter, but which are needed to meet Sandvik's requirements. They are presented in table 4.4.

Table 4.4. *Other components of COSMOS FX*

Component	***Relates to requirement***	***Decision***
Housing / protection	R3	Commercial metal housing / Components encapsulated in resin
Fastening	R4	Magnetic
LED lights	R13	6 LED lights
Teaching buttons	R10	4 buttons button pad
Memory card / adaptor	R6	SD card + SD card adaptor for Teensy
Real time clock	R7	Teensy compatible real time clock with crystal

Housing must to meet the protection requirement R3. As COSMOS FX is used in rugged conditions, metallic housing provides the best protection for the instrument. Keskustekniikka could order purpose-built housing, but because the sales volume is unknown, using commercial housings is cheaper and more sensible. Most of the commercial housings are, however, less than IP67 protected, which is why the components will be encapsulated in resin in order to ensure adequate protection.

The requirement R4 requires fast and simple fastening. That is why COSMOS FX will be attached to a mining machine with magnetic fastening. It is a faster and simpler method than, for example, screw fastening.

The requirement R13 requests a LED light or lights to indicate the status of the COSMOS FX instrument. This requirement is met by adding 6 LED lights: 1 to indicate when the instrument is on, 1 to indicate when a mining machine is operating, and 4 to indicate different operation modes. For the protection class requirement R3, all the LED lights will be installed under plexiglass inside the housing.

The requirement R10 is related to the configuration, but as will be explained in Section 4.3.2, the configuration for operation mode detection is performed with teaching buttons. For this, COSMOS FX needs buttons, which also have to meet the protection class requirement R3.

The requirement R6 requires six months data storage capability. This requirement is met by adding a memory card and memory card adaptor to Teensy's SPI bus. For the requirement R7, a real time clock is added to the instrument.

4.2.6 Circuit Board and Design

According to this implementation plan, the circuit board of the COSMOS FX instrument is built from two parts: the Teensy board and an additional COSMOS FX shield that is mated to Teensy, as shown in Figure 4.11. The shield will be designed after final component decisions, but it will contain all the necessary components needed to meet the requirements set for it and to ensure proper functionality.

Because the instrument will be under continual mechanical vibration, it would be better if Surface Mount Device (SMD) components were used. Furthermore, by using SMD components, the instrument would be much smaller. However, because the sales volume cannot be predicted, it is better to use through-hole components, when prototyping and manufacturing of a few instruments are easier and cheaper. Later, if the sales increased, through-hole components could be changed to SMD components.

Figure 4.11 drafts the structure of the COSMOS FX instrument and represents the main parts of it. Because the final component decisions are not made in this thesis, the figure does not present any particular components and neither is on a scale. The figure gives a good overview of the content of the instrument and shows what COSMOS FX would look like if it was built according to the plan presented in this thesis.

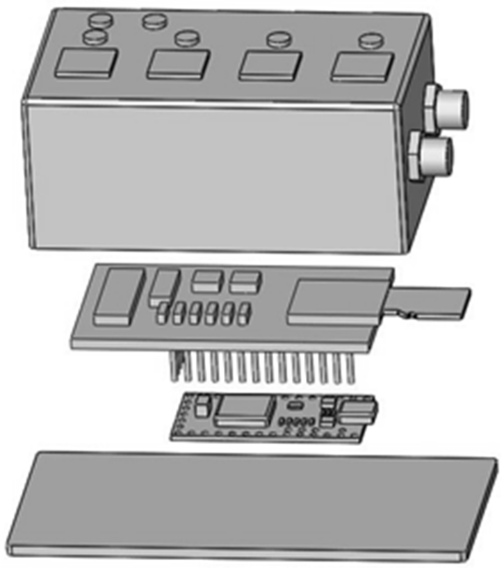

Figure 4.11. *General design of COSMOS FX*

4.3 Software of COSMOS FX

This section discusses the COSMOS FX software that is based on the existing COSMOS software. Because the aim of the thesis is not to represent the complete architecture of the software, the section introduces only new and use case specific features. These features are described at an abstract level, which means that no code is presented here.

4.3.1 General Features

Software is a part of the system that makes actual decisions about the information flow and performed actions. For that reason, almost all of the requirements must be taken into account when the software is built. In addition to the listed requirements, it also has to ensure that the software operates as Sandvik requires and as represented in system models in Section 4.1.3.

The requirement R5 requires that the instrument should work in the mining machines of all types. Although this thesis focuses only on a few type of drill rigs, the software is built so that it should also work in other types of machines. However, the testing of the other machine types is left for further development.

The requirement R8 requires automatic data transmission, which is implemented with WLAN as decided in Section 4.2.4. The automatic data transfer feature operates so that the collected data is transferred automatically to the COSMOS server/cloud whenever the COSMOS FX instrument is within WLAN coverage area. As mentioned in Section 4.2.4, the existing COSMOS software already has this feature, and thus no extra action is needed to take care of this feature.

The requirement R9 relates to the manual transmission. As mentioned in Section 4.2.4, the manual data transmission is implemented with USB, which means that the data must be automatically transferred to a USB flash drive when it is attached to COSMOS FX. This feature is added to the software: when the folder structure on the USB flash drive is set in a certain way, the software automatically transfers the content of the memory card to the USB flash drive.

The requirements R10, R11, and R12 address in the operation mode recognition feature and the configuration. As mentioned in Section 4.2.3, the operation mode detection is based on the vibration, which is measured with the vibration sensor and analyzed with the FFT algorithm. The software uses 256-point *arm_cfft_radix4_f32* FFT function, which transforms the vibration signal's time domain to the frequency domain that contains 256 frequency components. However, because the components 128-256 Hz are a mirror image of the components 1-127 Hz, the software observes only the components 1-127 Hz [5]. The more detailed description of the configuration and operation mode detection function is presented in the following sections.

The software writes all the relevant data to a log file, which is stored on the memory card. The log file contains measurement results that are saved every second and the operation mode information that is saved every 5 seconds. A new log file is started every 8 hours in order to limit the size of a single log file.

4.3.2 Configuration

The requirement R10 demands that the COSMOS FX instrument should not need any instrument-specific configuration. However, because the system must work in all types of mining machines (CR5) whose operation modes vary, the instrument needs some configuration for operation mode detection. This configuration is built with teaching buttons, which helps even an inexperienced person to configure the COSMOS FX instrument as the requirement R4 requires.

Figure 4.12 shows how the configuration for the operation mode detection is created. When a user presses a button, the teaching function measures and calculates 3D vectors of the frequency components 1-127 Hz. The components are measured and calculated five times at 1 second intervals in order to get 5 seconds averaged value for each component. After the calculations, the function finds the highest frequency components and stores them into a reference array. The created reference array is written in the configuration file and this way it can be used for operation mode detection. The detection function will be explained in the next section.

When the mode 1 is taught, the teaching function also calculates a reference value, which is used to recognize an idle mode of a mining machine. The reference value is calculated by taking the average value of all measured frequency components and divided it by two.

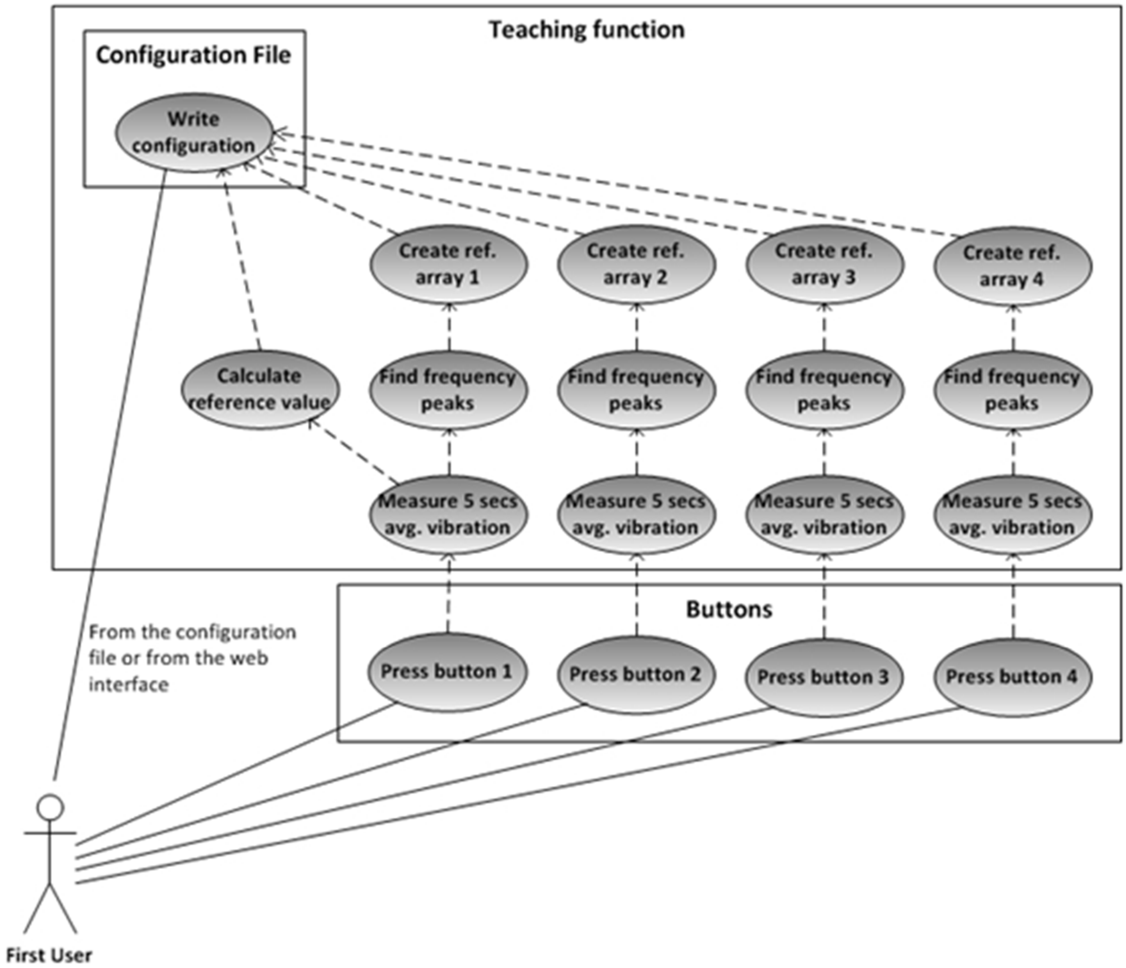

Figure 4.12. Use case / Process Model for configuration

For example, if Table 4.5 presents the averaged measurement results of 5 measurements and if the size of the reference array is 3, the reference array gets the frequency components 15 Hz, 43 Hz, and 73 Hz, assuming that values of the unlisted frequency components are lower than 53.77. The reference value for idle mode recognition, in turn, is (unlisted values are not included):

$$reference\ value = \frac{26.99}{2} \approx 13.50.$$

Table 4.5. *Example of averaged frequency components*

Frequency Component	***x***	***y***	***z***	***Vector***
1	1.26	1.54	0.93	2.20
2	0.80	0.33	0.59	1.05
3	1.90	1.13	1.69	2.78
4	3.11	2.27	0.74	3.92
5	3.45	2.12	1.21	4.23
…	…	…	…	…
15	27.32	39.38	17.01	**53.77**
…	…	…	…	…
43	57.93	89.10	24.38	**109.04**
…	…	…	…	…
73	31.49	42.20	18.56	**55.83**
…	…	…	…	…
86	9.61	22.97	7.84	26.10
…	…	…	…	…
127	4.59	9.68	2.15	10.93
avg.	14.15	21.07	7.51	26.99

In addition to the operation mode configuration, COSMOS FX also needs a unique identification (ID) and network configuration. These configurations are implemented so that by default, network settings come from a Dynamic Host Configuration Protocol (DHCP) server, if available, and the ID of an instrument is set according to the Media Access Control (MAC) address of a Teensy board.

All configurations can be edited from the configuration file, which is located on a memory card. However, because Sandvik requires that the installation and configuration should be done in 15-30 minutes by an inexperienced person (CR4), a web interface that facilitates configuring will be also created. Building the web interface is left for further development.

4.3.3 Operation Mode Detection

Figure 4.13 displays the operation principle of the operation mode detection function. Also, this function measures and calculates 3D vectors of frequency components 1-127 Hz. At first, the function confirms that the average value of all components is greater than the reference value that indicates the idle mode. After that, if the average value is greater than the reference value, the function calculates a comparison value for each operation mode. The comparison value is calculated by summing up the values of frequency components that are around each reference components (frequency components in the configuration file). The highest comparison value reveals the current operation mode.

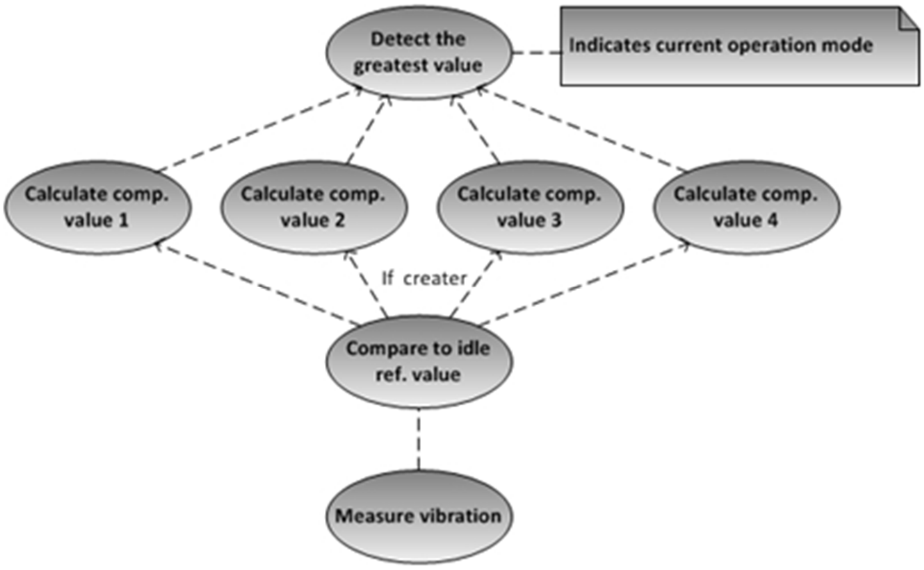

Figure 4.13. *Process Model for operation mode detection function*

For example, if the operation mode configuration is the configuration presented in Table 4.6, the measurement results are the results presented in Table 4.7, and the frequency range is 2, the function operates as follows:

At first, the function confirms that the average value of all frequency components is greater than 13.50. Because 26.84 is greater than 13.50, the function calculates the comparison values by calculating the sums of the frequency components 41-45 Hz, 71-75 Hz, and 13-17 Hz and 125-127 Hz, 84-88 Hz, and 3-7 Hz. What need to be taken into consideration is that at the both ends of the frequency component spectrum, the function sums also mirrored values. For instance, in this example where the reference component is 127 Hz, the application sums values of 125, 126, 127, 126, and 125 Hz.

In this example, the comparison values are:

$comparison\ value\ 1 = 651.05$ and

$comparison\ value\ 2 = 122.32$.

The comparison value 1 is greater, which means that according to the function, the current operation mode is Mode 1. Because the instantaneous values may vary fast, a function uses a few seconds hysteresis in order to prevent unwanted rapid operation mode changes.

Table 4.6. *Example of reference arrays*

Peak	***Mode 1***	***Mode 2***
Peak 1	43	127
Peak 2	73	86
Peak 3	15	5
Reference value	13.50	

Table 4.7. *Example of frequency components*

Frequency Component	***x***	***y***	***z***	***Vector***
3	1.90	1.13	1.69	2.78
4	3.11	2.27	0.74	3.92
5	3.45	2.12	1.21	4.23
6	5.24	6.78	2.08	8.82
7	4.45	5.22	1.88	7.11
…	…	…	…	…
13	24.97	33.35	14.89	44.24
14	25.27	34.45	15.12	45.32
15	29.45	41.56	18.07	54.05
16	24.41	35.55	14.56	45.52
17	20.11	29.67	12.21	37.87
…	…	…	…	…
41	23.89	40.00	12.13	48.14
42	28.56	42.45	15.78	53.54
43	58.94	91.23	25.89	111.66
44	9.89	12.45	5.48	16.82
45	4.86	4.62	2.37	7.11
…	…	…	…	…
71	5.89	7.88	3.84	10.56
72	12.84	15.57	8.85	22.04
73	34.61	43.82	19.78	59.24
74	35.89	46.67	23.94	63.56
75	21.38	19.89	11.53	31.40
…	…	…	…	…
84	2.41	1.44	2.15	3.53
85	3.95	2.88	0.94	4.98
86	4.38	2.69	1.54	5.37
87	6.65	8.61	2.64	11.20
88	2.25	1.87	0.57	2.98
…	…	…	…	…
125	8.35	6.09	1.99	10.52
126	9.26	5.69	3.25	11.34
127	14.06	18.20	5.58	23.67
avg.	15.37	20.15	8.24	26.84

4.4 Database and Reporting System

The database and reporting system can be run either on a local server or in the cloud. Even though both of them are beyond the scope of this thesis, this section introduces a few core features of them.

First of all, because COSMOS product family already contains the COSMOS Cloud service, the most practical solution is to locate the database and reporting system into the cloud. In this way, the optional requirement R15 is also met.

As mentioned in the previous sections, COSMOS FX automatically transfers data to the database when it is within WLAN coverage area. The COSMOS software has the capability to send data directly to COSMOS Cloud over Hypertext Transfer Protocol Secure (HTTPS) protocol, but some customers might have restrictions when it comes the Internet connection in their networks. For that reason, it is best to add a connection point computer, which routes the networks and transfers the data received from COSMOS FX to COSMOS Cloud. This connection point computer can also be used for manual data transfer.

The reporting system should generate clear graphical reports that represent usage information, as the requirement R16 requests. For that, the system needs a cloud-based reporting system that presents the information and generates reports that show the information from selected time periods. Development of this system is left for further development to be done in the future.

5. REALIZATION OF THE INSTRUMENT

This chapter discusses the realization of the data collecting instrument. The first section introduces the first prototype of COSMOS FX. The second section represents the test results of the operation mode detection tests. In the last section, the features of the designed instrument are evaluated on the basis of the requirements set up for the instrument and the system.

All the results presented in this chapter are based on the assumption that the upcoming commercial instrument will be built according to the design plan made in the previous chapter. However, the final design might still undergo some changes. For example, Texas Instruments has a single-chip microcontroller unit CC3200, which contains built-in WLAN connectivity [28]. Although Teensy is an excellent platform for the product concept and software testing, Texas Instruments' unit provides an interesting alternative to Teensy. The unit should run the developed COSMOS software. Also, by using this unit, the instrument would not need an additional module for WLAN, which would make it simpler and smaller. Texas Instruments' unit is not covered in this thesis because it became an option only at the final stages of this thesis project. For that reason, it was left for further development.

5.1 First Prototype

Figure 5.1 shows the first prototype of COSMOS FX. It was built for software testing, which is why the features and the appearance of the instrument deviated from the design choices made in the previous chapter. For example:

1. The prototype did not contain the shield board as represented in Section 4.2.6.
   → The components were development components, which were connected with wires.
2. The instrument did not contain a power connector nor an additional connector.
   → The prototype was battery-powered.
3. The instrument did not contain a WLAN module.
4. The case was made of plastic.
5. The LED lights, the buttons, and the housing were not IP 67 sealed.
6. The instrument did not have magnets for fastening.

Figure 5.1. *The first prototype of COSMOS FX*

Despite the deviations, the prototype reflected the instrument designed in the previous chapter: its physical properties were much weaker, but it contained similar elements and features that were designed. Hence, the prototype could be regarded as a sample, which shows how the commercial instrument would look and operate if it was built according to the plan made in this thesis.

5.2 Operation Mode Detection Test

The operation mode detection tests were divided into two parts. The first shorter test ensured that the software worked and the operation modes of the underground drill rig could be detected. The second test compared the collected usage data with the actual usage of the mining machine.

5.2.1 First test

The first test was performed with the underground tunneling jumbo drill rig DT820. The prototype was attached to the frame of the drill rig so that the vibration sensor inside the instrument could measure the vibration of the machine. The aim of the test was to observe how the software tested in an office environment operated in the mining machines of Sandvik and to ensure that the software could detect the different operation modes and the changes between them.

At first, the operation mode detection feature was tested by taking two frequency components into the reference array. However, the test revealed that the results improved when the number of the components was increased to three. The hertz range used throughout the test was 2 Hz.

Table 5.1 shows the reference arrays that were created by teaching three different operation modes: the powerpack on mode, the diesel engine on mode, and the drilling mode.

Table 5.1. *Reference arrays of the DT820*

Mode	*Peak 1*	*Peak 2*	*Peak 3*
Powerpack on	25 Hz	31 Hz	76 Hz
Diesel engine on	23 Hz	24 Hz	47 Hz
Drilling	30 Hz	60 Hz	45 Hz
Reference value		4.9	

When the reference arrays are compared with the figures 5.2-5.4, which represent the frequency domains measured during the tested operation modes, it can be seen that they correlate to each other well. Also, the instant observations about the operation mode recognition were promising during the test. In some cases though, the teaching had to be performed a couple of times before the operation mode changes were detected as they were supposed to. Another note was that the application detected the diesel engine mode better if the mode was taught at higher rpms than the idle rpm of the diesel engine.

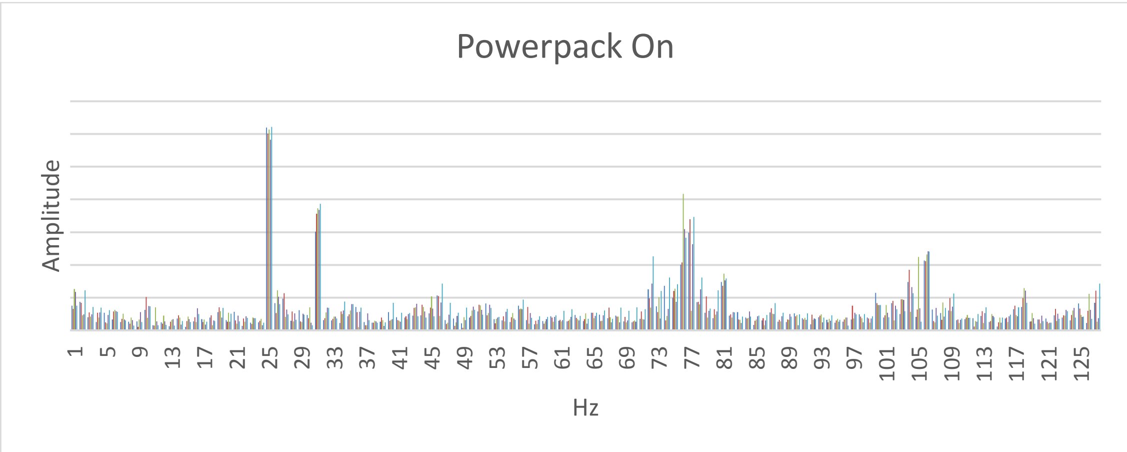

Figure 5.2. *Frequency domain: DT820 powerpack on*

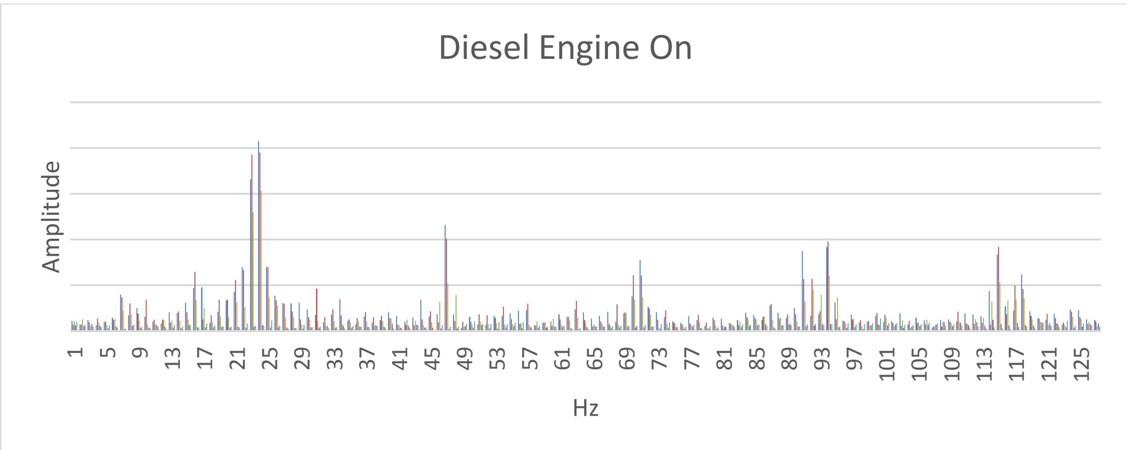

Figure 5.3. *Frequency domain: DT820 diesel engine on*

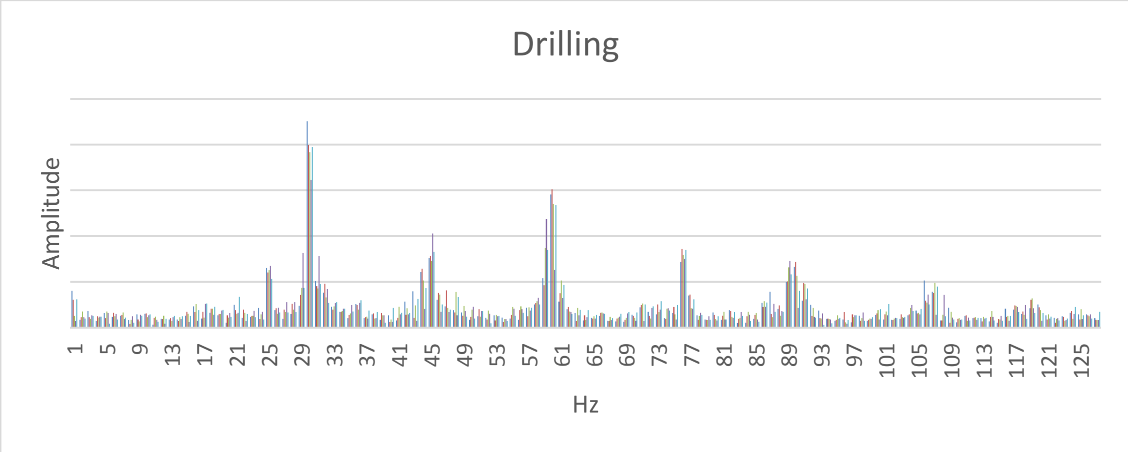

Figure 5.4. *Frequency domain: DT820 drilling*

Figure 5.5 shows the timeline of the operation modes in the first test. Because the aim of the test was to ensure that the software detected operation modes and their changes, the operation modes were changed more frequently than in the normal use. For that reason, the modes in the timeline are bouncing from one mode to another.

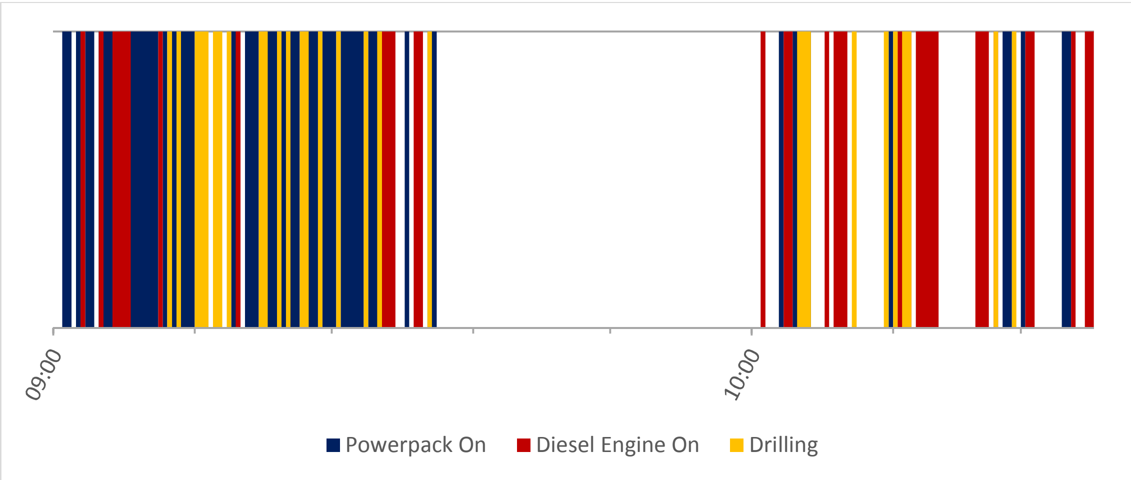

Figure 5.5. *Timeline of operation modes in the first test*

Analyzing the timeline revealed a few issues. The first was that the operation mode data was not recorded every 5 seconds as it should have been. The missing data points can be seen as small gaps between 9:00-9:30. By analyzing the log file, it was noticed that the issue might relate to operation mode teaching, which means that it is not an issue during the normal use. The second issue was that sometimes the operation mode detection function detected the movements of a boom as drilling. This phenomenon can be seen, for example, between 9:15-9:30 where the short drilling periods occur among the powerpack on mode. This issue may be solved by optimizing configurations, such as hertz range and hysteresis when the rapid operation mode changes do not affect the results.

5.2.2 Second test

The second test was performed with the underground production drill rig DL421. The aim of the test was to use the drill rig as it is usually used and compare the collected data with actual production events. As in the first test, the number of measured frequency peaks was 3 and the frequency range was 2 Hz.

The most significant test results are presented in the following tables and figures. Table 5.2 shows the reference arrays that were created by teaching three different operation modes: powerpack on, diesel engine on, and drilling. As can be seen, the frequency peaks with the DL421 were different than with the DT820. It supports the idea of the teaching feature because each machine type will require a machine-specific configuration as was assumed at the beginning of the test.

Table 5.2. *Reference arrays of the DL421*

Mode	*Peak 1*	*Peak 2*	*Peak 3*
Powerpack on	77 Hz	78 Hz	65 Hz
Diesel engine on	86 Hz	44 Hz	23 Hz
Drilling	91 Hz	32 Hz	77 Hz
Reference value		5.16	

Table 5.3 presents the actual production events in the second test. When the events are compared with Figure 5.6, which represents the timeline of the operation modes in the test, the similarity between them can be seen. It should be noticed that the results were not completely accurate. The inaccuracies will be analyzed next.

Table 5.3. *Event history in the second test*

Time	*Event*	*Note*
10:50	Test started	Operation modes were taught before starting.
10:52	Powerpack turned on	
10:53	Drilling of the first hole started	Every 2:15 minutes a new rod was attached to the drill string. Attaching took app. 30 seconds.
11:11	First hole drilled	The rods were detached from the drill string.
11:22	Drilling of the second hole started	Every 2:15 minutes a new rod was attached.
11:39	Second hole drilled	The rods were detached.
11:46	Machine turned off	
11:53	Diesel engine started (rpms ranging)	The instrument was not able to detect the engine mode. The operation mode had to be taught again.
11:58	Diesel engine running (idle rpms)	
12:05	Machine turned off	
12:10	Test ended	

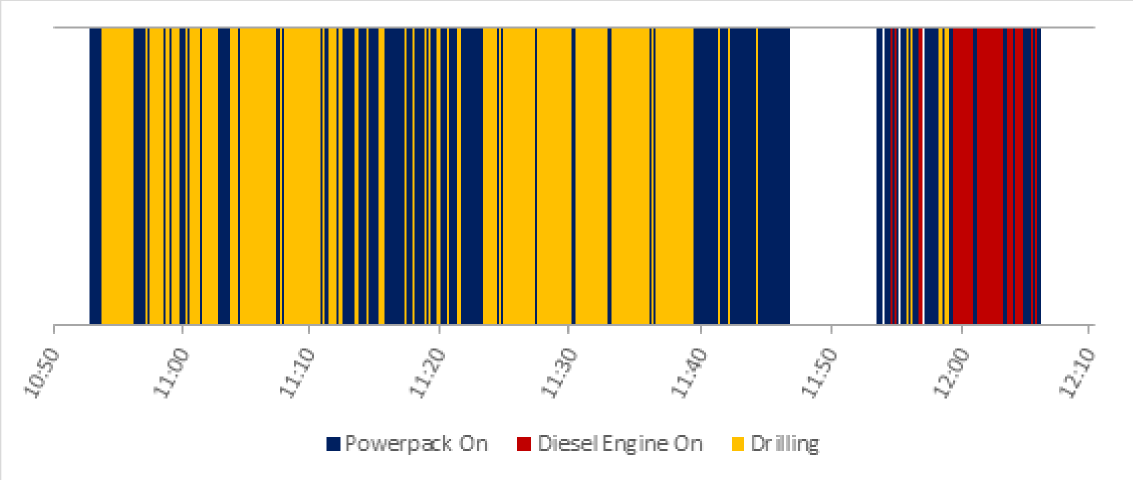

Figure 5.6. *Timeline of operation modes in the second test*

The first inaccuracy was related to the nature of the drilling process. The test was performed with the long-hole drill rig, which drilled approximately couple meters at a time, after which a new rod was attached to the drill string. The attaching phase caused approximately 30 seconds interruption every 2:15 minutes, which can be also seen from the timeline. Also, the actual drilling process needed a few breaks when the drill bit or sting was jamming or if there was some other reason why the drilling did not progress as it should have been. Breaks of this kind can be seen, for example, at times 10:56 and 11:03.

The second inaccuracy was related to the use of the hydraulics. As observed already in the first test, sometimes the movements of the boom were detected as drilling. This phenomenon occurred, for example, when the drill rods were detached from the drill string. In the timeline, this can be seen between the time periods 11:11-11:22 and 11:39-11:46. However, it has to be noted that sometimes the detaching also needed a short percussion period so that the drill rod could be detached. For that reason, all of the drilling periods during the detaching phase shown in the timeline are not measurement errors.

The greatest inaccuracy in the test occurred when the diesel engine was started. Although the instrument was able to detect the diesel engine when the operation modes were taught, it did not detect it in the test phase. For that reason, the diesel engine mode was taught a couple of times again during the test after which the instrument was able to detect the operation mode. The inaccuracy indicates that the teaching should be performed carefully and with higher rpms than the idle rpm of the engine.

If the timeline is simplified, it can be represented as shown in Figure 5.7. The simplified timeline is obtained by removing the distortions and other rapid operation mode changes from the measurement results. As can be seen, the modified timeline correlates perfectly with the actual production events except in the time period 11:53-11:59, which is colored as gray. The similarity between the actual events and the simplified timeline proves that by resolving the most significant inaccuracies, the results would be as hoped. This goes to show that the instrument could be able to function as expected.

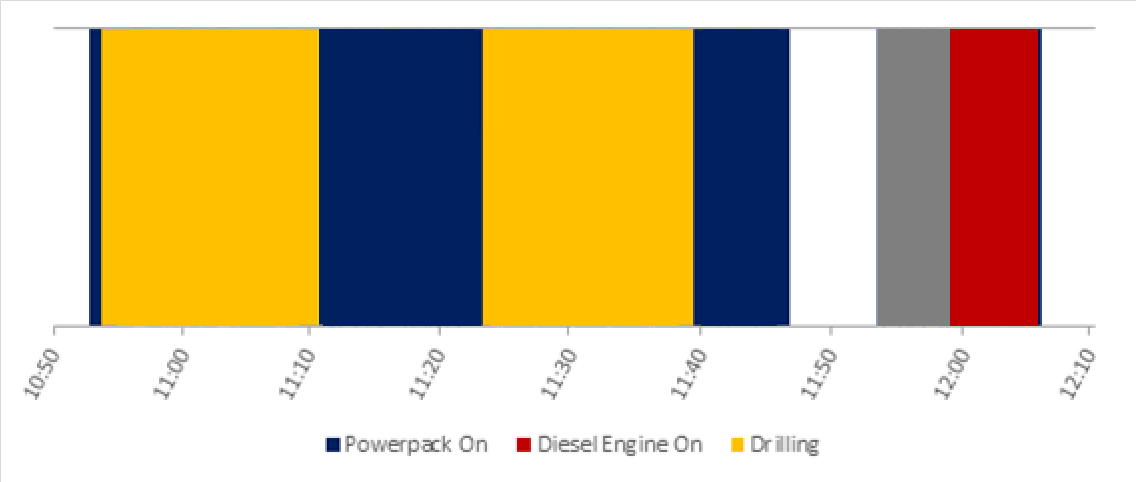

Figure 5.7. *Simplified timeline of operation modes in the second test*

The test results of the second test show that the software and its configuration still need some work and study. For example, it would be good to study:

1. Which is the best hertz range?
2. How many frequency peaks should be measured?
3. How many seconds hysteresis produce the most accurate results?
4. Does a shorter or longer teaching time produce better results?

Also, it would be good to study the optimal rpm for the teaching so that the detection problems can be avoided in the future.

5.2.3 Test Summary

On the basis of the test results, it is possible to review how the COSMOS FX instrument would be suitable for Sandvik's machines. The suitability depends on Sandvik's wishes: what they want to measure and how accurate data they want to obtain.

For example, when measuring the drilling mode, does Sandvik want to measure the actual drilling/percussing time or the total operating time, including breaks and rod attaching phases? If they want to measure the actual drilling time, the COSMOS FX instrument works as it is and needs only a little optimization; but if they want to include all interruptions and measure the total operating time, the instrument also requires some supporting measurements. For that, the instrument should also observe some relay of the machine or measure some other variable.

When talking about the accuracy of the measurements, the results show that the instrument cannot measure completely accurate operation times. It means that if Sandvik wants to obtain the accurate operation times, the vibration measurement is not the best option. If the accuracy, in turn, is not one of the most significant factors and the indicative operation times are acceptable, the vibration measurement is a possible solution.

All in all, the results show the potentiality of the COSMOS FX instrument. Although there are still certain issues and inaccuracies, the results encourage continuing the development. With a little optimization and development, the greatest issues can probably be resolved. It is also worth of noting that although the instrument would not work in Sandvik’s machines, there are still several other potential target machines and facilities where the COSMOS FX instrument can be used.

5.3 Fulfillment of the Requirements

Table 5.4 presents information about how the designed instrument meets Sandvik’s requirements. The fulfillment level is divided into four categories:

1. The prototype met the requirement.
2. The upcoming commercial instrument will meet the requirement if it is built according to design choices made in Chapter 4.
3. The fulfillment of the requirement is left for further development.
4. The requirement cannot be fulfilled.

Each requirement is evaluated and its fulfillment level is indicated by a color-coded indicator. Some of the requirements are placed under multiple categories if they have already been met in the prototype or if the designed instrument will only partially meet them.

Table 5.4. *Fulfillment of Sandvik's requirements*

ID	***Requirement description***	***M/O***	***Fulfillment***	***Remark***
R1	9-36 VDC power supply	M		
R2	Overvoltage protection	M		
R3	IP67 sealed	M		
R4	Installation in 15-30 minutes	M		The magnetic fastening and the teaching buttons enable fast installation
R5	Works in machines of all types	M		The software should work in all machine types, but was tested only with drill rigs DT820 and DL421. Other machine types are left for further testing.
R6	6 months data storage	M		A memory card enables 6 months data storage.
R7	Real time clock	M		
R8	Automatic data transfer	M		The instrument and software support automatic data transfer over WLAN. Receiving database and other technology options are left for further development.
R9	Manual data transfer possibility	M		The data is transferred to a USB flash drive when the folder structure of the attached drive is set in a certain way.
R10	Simple and easy configuration	M		The teaching buttons enable fast and easy configuration. The web interface is left for further development.
R11	Detect idle, engine on, and operating modes	M		The modes can generally be detected in DT820 and DL421. Optimization and testing of the other machine types are left for further development.
R12	Detect power pack on mode and which boom is drilling	O		The powerpack on mode OK in DT820 and DL421. Detection of the boom is not possible. Testing of the other machines is left for further development.
R13	LED lights	O		6 LED lights indicate the instrument's status and a detected operation mode.
R14	Possibility for external sensors	O		M12 inputs enable external sensors. The software must be built for each case
R15	Connection to the cloud	O		The cloud-based database and reporting system are left for further development.
R16	Reporting system	O		The cloud-based reporting system is left for further development.

The prototype fulfills the req. The upcoming instrument will fulfill the req. The req. left for further development The req. cannot be fulfilled

As can be seen from Table 5.4, the first prototype met only a few requirements. However, the first prototype was built for software testing, and for that reason, it is more remarkable to review the requirements that will be met if the upcoming commercial instrument is built as designed in this thesis. As can be noted, the designed instrument will fully meet the requirements R1, R2, R3, R4, R6, R7, R9, and R13. The requirements R10, R11, R12, and R14, in turn, will be only partially met.

The requirement R10 requires that the configuration should be easy to perform. With teaching buttons, the operation mode configuration is easy and can be performed by anyone. Moreover, the planned web interface will provide an alternative way to perform the configuration. Because the building of the web interface is left for further development, the requirement is partly placed under the "*left for further development*" category.

The requirement R11 requires that the instrument should record the idle, the engine on, and the operating hours of a mining machine. As the test results show, the prototype was generally able to detect the diesel engine on mode and the drilling (operating) mode. However, the tests were performed only with underground drill rigs DT820 and DL421 whose operating mode and diesel engine mode are easily distinguishable. The measurement results might have changed if the tests had been performed with an electrical machine or with a loader whose operation modes do not cause a steady vibration. The test results also show that the software and its configuration still need optimization for more accurate measurements. The optimization and testing of the other machine types are left for further development.

The requirement R12 requests the instrument to record the powerpack on hours and to recognize which boom is drilling. According to the test results, the prototype was mostly able to detect the powerpack on mode of the tested drill rigs. However, as shown in the first test, it is not possible to recognize which boom is drilling. This is why the requirement R12 is partly under "*not realized*" category. Also, this requirement needs optimization and further testing.

The requirement R14 requests a possibility for external sensors. The additional M12 connector whose pins are connected to Teensy's inputs meets this requirement. Because a measurand depends on a connected unit, the software must be built separately for each case. This is the reason the requirement is partly under "*left for further development*" category.

The requirement R5 requires the instrument to operate in all Sandvik machines. COSMOS FX was designed so that it can be installed in all machine types of Sandvik and it should also detect different operation modes of those machines. In this thesis, the instrument was tested only with the underground drill rigs DT820 and DL421, which left the testing of the other machine types for further development.

The requirement R8 requires automatic data transfer to a server/cloud. The designed instrument will be capable of sending data automatically over WLAN, but because the receiving database is left for further development, this requirement cannot be considered fulfilled. In addition to WLAN, it would be good to consider other technology options as well.

The requirement R15 requests a connection to a cloud in the future whereas the requirement R16 asks for a reporting system. These requirements are not related to the COSMOS FX instrument, which is the core part of this thesis. For that reason, the requirements are left for further development.

6. CONCLUSION

This thesis studied ways to adapt Keskustekniikka's COSMOS data collection system to Sandvik's mining machines. The key objective of the thesis was to design a product concept for the intelligent data collection instrument – COSMOS FX, which collects usage data from a mining machine by measuring the vibration of the equipment. The instrument was designed especially for Sandvik's machines, but as a part of Keskustekniikka's COSMOS product family, it had to be suitable for other types of machines and facilities as well.

First, the most useful system models were modeled according to the IoT Architectural Reference Model. This reference model is the result of the IoT-A project and provides a common foundation for designing IoT systems. The model is composed of three parts: the IoT Reference Model, the IoT Reference Architecture, and the guidelines. These parts, in turn, consist of several sub-models, views, and best practice guides that facilitate and guide the designing process of an IoT system.

The design concept of the instrument was divided into two parts: the hardware and the software design. The hardware of the instrument was designed at a general level. This means that no final components were selected, but the most significant hardware decisions were made. Those decisions outline the hardware options and ensure that the upcoming instrument will meet Sandvik's requirements. The software part, in turn, was focused on the operation mode detection feature. It is a new and the most significant feature of COSMOS FX while many other required features were already included in the existing COSMOS software.

The designed operation mode detection feature was tested with the prototype instrument in the underground drill rigs DT820 and DL421. The results show that in most cases, it is possible to detect the required operation modes, which supports the idea of using a vibration sensor for operation mode detection. However, there were still many measurement errors; therefore, the software and especially its configuration still need to be adjusted. The test results also show that the operation modes must have a clear difference in vibrations. This might cause problems, for example, in loading and hauling machines. In those machines, the difference between "operating" and "engine on" mode is small and is caused only by the movements of the machine, which does not cause steady vibration.

On the basis of the designed instrument and the software tests, it possible to estimate how the instrument fulfills Sandvik's requirements. Because the prototype was built for the software tests, it met only 3/3/16 requirements (3 fully and 3 partially met requirements out of 16 requirements). More interesting for this thesis is the estimation that the upcoming commercial COSMOS FX instrument will meet 8/4/16 requirements if it is built according to the design choices made in this thesis. When taking into account that 3 out of those 4 unmet requirements were related to the features that were beyond the scope of this thesis, the results could be considered good.

Although the presented results seem good, it has to be noted that there were a few measurement errors, which have to be resolved before the instrument can be used for commercial purposes. It is also worth noting that some unexpected issues might still occur, which would mean that the expected results cannot be achieved. Despite the certain issues and inaccuracies, the test results encourage continuing the development of the instrument. By resolving the most critical issues, the instrument could be sold to Sandvik as was originally planned. It is also worth noting that in addition to Sandvik, there are several other potential target customers, who might want to buy the instrument. This means that even if Sandvik would reject the instrument, the COSMOS FX instrument is still worth further development and will be placed into commercial distribution.

REFERENCES

[1] Arduino, Arduino Uno - Technical Specs, Arduino, website. Available (accessed on 15.11.2015): https://www.arduino.cc/en/Main/ArduinoBoardUno.

[2] Ashton K., That 'Internet of Things' Thing, RFID Journal, Jun 22, 2009.

[3] Bluetooth SIG, Competing Technologies, Bluetooth SIG, website. Available (accessed on 11.5.2015): http://www.bluetooth.com/Pages/Competing-Tech.aspx.

[4] Chaouchi H., Introduction to the Internet of Things, in: The Internet of Things: Connecting Objects, 1st ed., ISTE Ltd and John Wiley & Sons, Inc., Great Britain and the United States, 2010, pp. 1-34.

[5] CMSIS, Complex FFT Functions, CMSIS - Cortex Microcontroller Software Interface Standard, website. Available (accessed on 05.12.2015): http://www.keil.com/pack/doc/cmsis/dsp/html/group___complex_f_f_t.html#ga521f670cd9c571bc61aff9bec89f4c26.

[6] Coley G., BeagleBone Black System Reference Manual, Rev A6. 2013, 124 p.

[7] Control Express Finland Oy, WBS-Remote monitoring solutions, Control Express Finland Oy, website. Available (accessed on 24.10.2015): http://www.cef.fi/en/wbs-remote-monitoring-solutions/.

[8] ETH Zurich, Internet of Things 2008: International Conference for Industry and Academia, ETH Zurich, website. Available (accessed on 29.6.2015): http://www.iot-conference.org/iot2008/.

[9] Google Trends, Google trends for Internet of Things and IoT, Google, website. Available (accessed on 29.6.2015): https://www.google.fi/trends/explore#q=internet%20of%20things%2C%20IoT&cmpt=q&tz=Etc%2FGMT-3.

[10] Gratton D.A., Near Field Communications: The Smart Choice for Enabling Connectivity, in: Developing Practical Wireless Applications, 1st ed., Elsevier, 2007, pp. 216-224.

[11] Hanhimäki T., Industrial Internet Cosmos by CLS, CLS Engineering Oy, Presentation, 2015, 9 p.

[12] Hanhimäki T., OPC to CLOUD, CLS Engineering Oy, Presentation, 2015, 10 p.

[13] Höller J., Tisatsis V., Mulligan C., Avesand S., Karnouskos S., Boyle D., From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence, 1st ed. Academic Press, UK, 2014, 352 p.

[14] Industrial Internet Consortium, Industrial Internet Reference Architecture, Industrial Internet Consortium, website. Available (accessed on 27.7.2015): http://www.iiconsortium.org/IIRA.htm.

[15] IoT - A, Deliverable D1.5 – Final architectural reference model for the IoT v3.0, IoT-A, 2013, 482 p.

[16] IoT-A, Introduction to the Architectural Reference Model for the Internet of Things, 7th ed. IoT-A, 2013, 26 p.

[17] Keskustekniikka, Cosmos Data Collection System: Real-time information on the operating condition of machinery, Keskustekniikka, website. Available (accessed on 4.5.2015): http://www.keskustekniikka.fi/en/index.php/products/cosmos-data-collection-system.

[18] Keskustekniikka, COSMOS: Enable Industrial Internet, Keskustekniikka, website. Available (accessed on 10.09.2015): http://cosmos.keskustekniikka.fi/.

[19] Keskustekniikka, X10 Startup and End User Manual, v.1.0. 29 p.

[20] Kivimaa T., Jokela T., COSMOS X10 User Manual, v.1.0. CLS-Engineering, 2015, 9 p.

[21] MIT Technology Review, Business Report - The Internet of Things, MIT Technology Review MIT, 2014, Available: http://www.technologyreview.com/news/527356/business-adapts-to-a-new-style-of-computer/.

[22] Object Management Group, Introduction To OMG's Unified Modeling Language® (UML®), Object Management Group, website. Available (accessed on 09.07.2015): http://www.uml.org/.

[23] PJRC, Teensy 3.1 Technical Specifications, PJRC, website. Available (accessed on 12.11.2015): http://www.pjrc.com/teensy/teensy31.html.

[24] Raspberry Pi, Raspberry Pi 2, Model B Datasheet, 2015.

[25] Sandvik Mining, Products Equipment, Sandvik Mining, website. Available (accessed on 26.6.2015): http://mining.sandvik.com/en/products/equipment.

[26] SparkFun Electronics, SparkFun Electronics Main Page, SparkFun Electronics, website. Available (accessed on 26.8.2015): https://www.sparkfun.com/.

[27] Svensberg V., Mannonen P., M.Sc, Manager, B.Sc, Product Manager, Automation Engineering, Sandvik Mining, Tampere, Customer meeting on 24.3.2015.

[28] Texas Instruments, CC3200 Datasheet, CC3200 SimpleLink™ Wi-Fi and Internet-of-Things solution, a Single-Chip Wireless MCU, rev. F. Texas Instruments, 2013, 71 p.

[29] Yepzon, Yepzon Quick-Guide, 13th ed. Yepzon Group.

APPENDIX A: FFT ANALYSIS

The Fast Fourier Transform (FFT) is an algorithm that can be used for analyzing vibrations or sounds. This algorithm processes an electrical signal in real time and transforms its time domain to the frequency domain that shows the components of the vibration. This transform is presented in Figure A.1. [1]

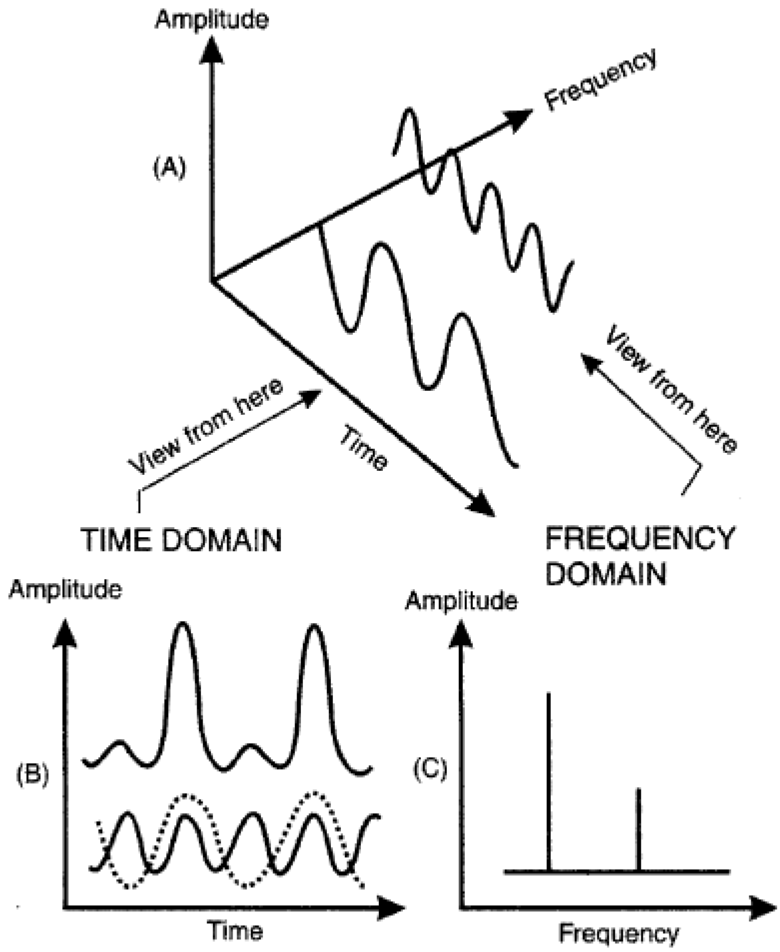

Figure A.1. *Fourier transform [1]*

The transform process starts from signal conversion changing an analog signal to a digital form. This conversion is performed by an analog-to-digital (A/D) converter that, first, samples an analog signal at a certain interval. This interval of the sampling is called a sampling rate, and it determines how fast samples are taken and how exact the signal replica is. According to the Nyquist theorem, the sampling rate has to be at least twice as high as the highest frequency component. After sampling, the analog values are converted to the digital form whose representation depends on the type of the A/D converter. For example, a 12-bit converter can represent the signal value in 4096 different levels and a 16-bit converter in 65536 levels. The number of different levels is known as the resolution of the converter. [1]

Next step of processing is windowing which is presented in Figure A.2. Its purpose is to reduce leakages that are caused when a signal is not periodic in the time window. At the top left, it is shown a signal that is periodic in the time window. Resulting FFT of the signal is shown at the bottom left. A signal that is not periodic in the time window is shown in the middle. It results in a leakage in the FFT as shown at the bottom center. When a window, at the top right, is applied, the leakage is reduced resulting in FFT shown at the bottom right. After windowing, a signal can be subjected to the FFT algorithm that gives the frequency domain of the signal. [2]

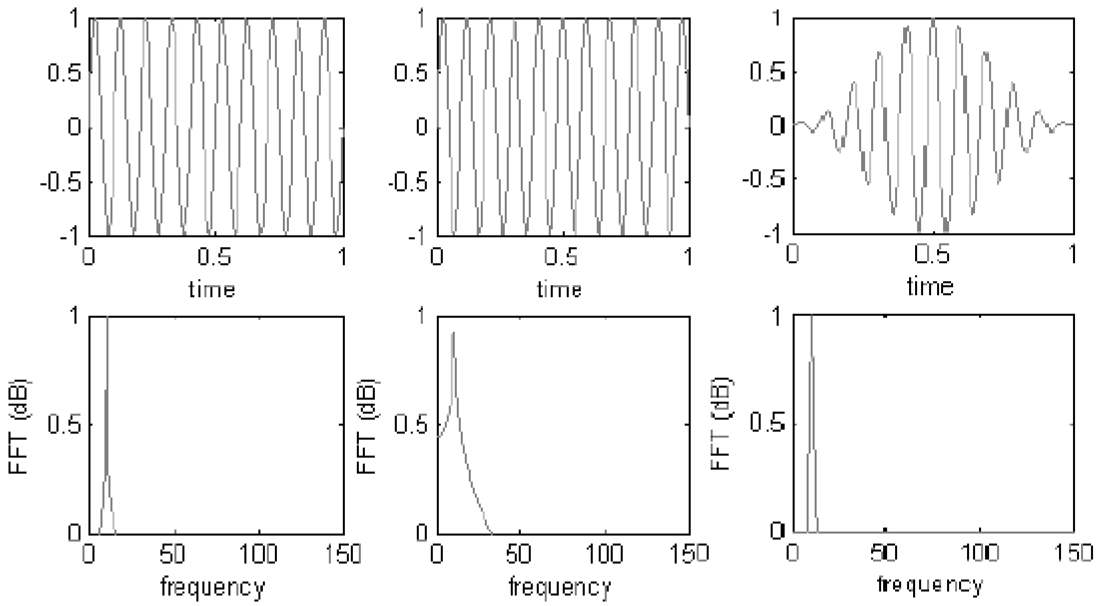

Figure A.2. *Signal windowing [2]*

REFERENCES:

[1] Girdhar P., Signal processing, applications and representations, in: C. Scheffer (ed.), Practical Machinery Vibration Analysis and Predictive Maintenance, 1st ed., Elsevier, 2004, pp. 55-88.

[2] Application Note ANO14: Understanding FFT Windows, LDS, 2003.

APPENDIX B: PRELIMINARY TEST

The preliminary test was performed with the early prototype of the data collection instrument, which included LSM303DLHC 3D accelerometer. The aim of the test was to confirm that the frequencies of the different operation modes differentiate so that they can be used for operation mode detection. The test also had to confirm that the accuracy of the sensor is good enough.

The test was carried out with three different types of mining machine: the underground drill rig DT820, the underground loader LH410, and the surface drill rig Pantera DI6400. Tables B.1 and B.2 present the strongest frequency peaks of the tested operation modes of DT820 and LH410. The results of Pantera DI6400 are not presented because the machine was not fully operational when it was tested. The frequency domains of the tested operation modes are presented in Figures B.1-B.8. Figure B9 shows the instrument used in the test.

The main findings of the test were:

1. Frequency components can be distinguished from the vibration and the operation modes can be detected
2. The instrument can be located wherever on the machine's frame because the frequency components are always same.
3. It is not possible to detect which boom is drilling.

Table B.1. *Test results of the underground drill rig DT820*

Machine Type: DT820		**Test Date:** 28.04.2015
Operation Mode	**Strongest Frequency Peaks (Hz)**	
	Location 1	**Location 2**
Power Pack On (Boom 1)	20-30 Hz / 50-60 Hz / 70-80 Hz	20-30 Hz / 50-60 Hz / 70-80 Hz
Drilling	50-60 Hz	50-60 Hz
Diesel idle	20-30 Hz	20-30 Hz

Table B.2. *Test results of the underground loader LH410*

Machine Type: LH410		**Test Date:** 28.04.2015
Operation Mode	**Strongest Frequency Peaks (Hz)**	
	Location 1	**Location 2**
Operating (bucket only)	0-10 Hz / 30-40 Hz / 70-80 Hz	-
Diesel idle	20-30 Hz	-

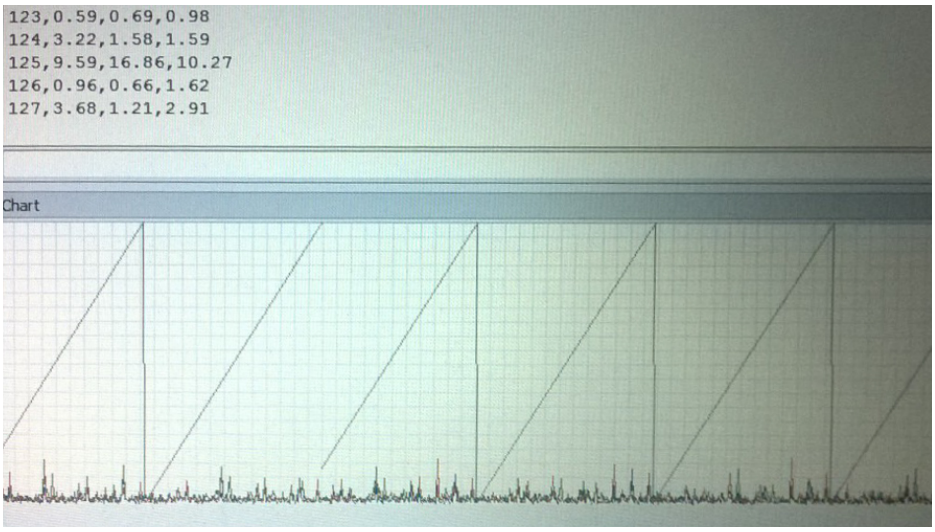

Figure B.1. *DT820 power pack 1 on (location 1)*

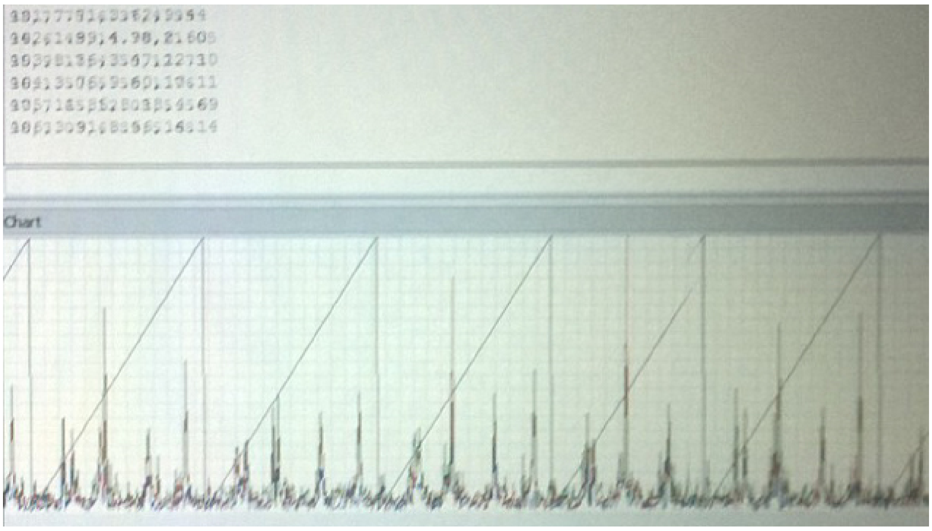

Figure B.2. *DT820 boom 1 drilling (location 1)*

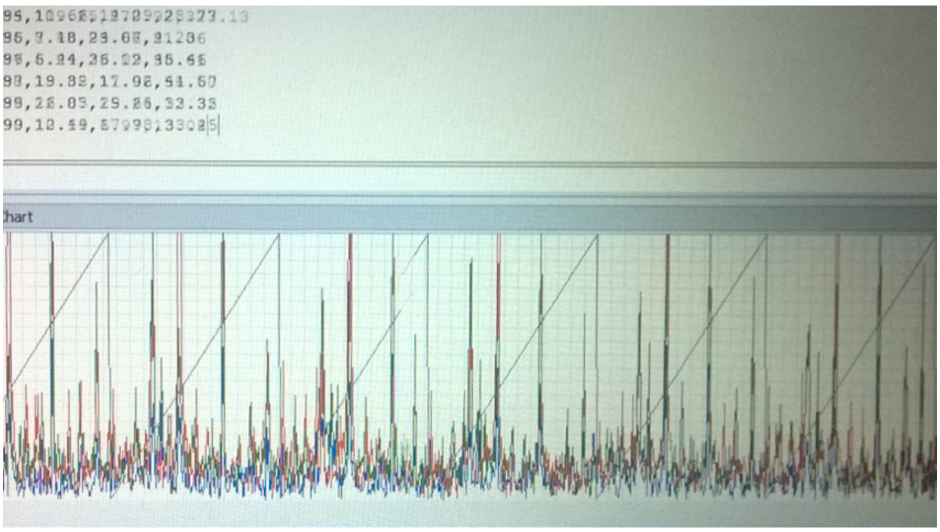

Figure B.3. *DT820 boom 1 drilling (loose mounting)*

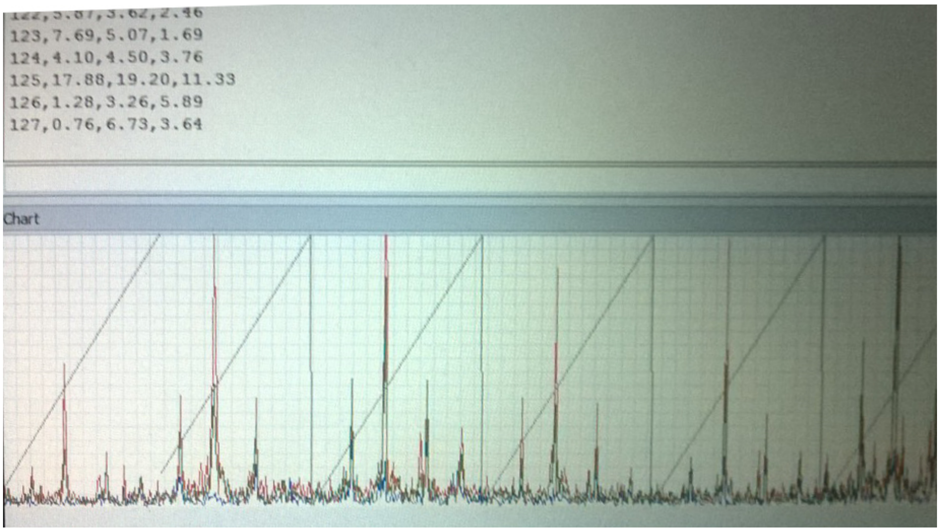

Figure B.4. *DT820 boom 1 drilling (location 2)*

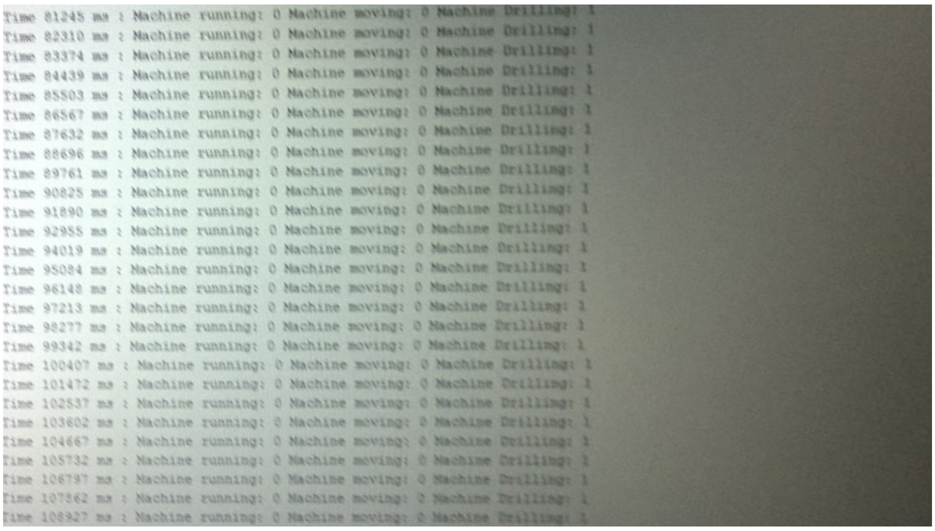

Figure B.5. *DT820 test of operation mode detection program*

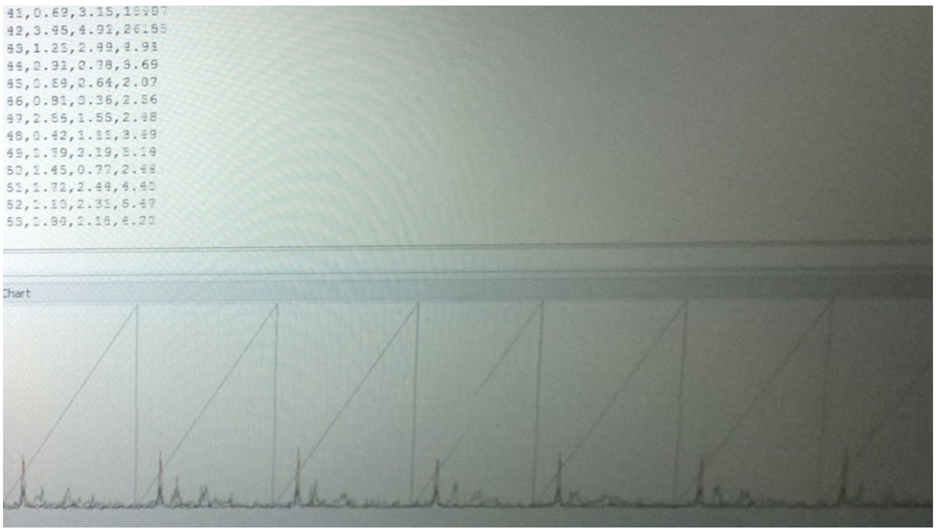

Figure B.6. *DT820 diesel engine idle*

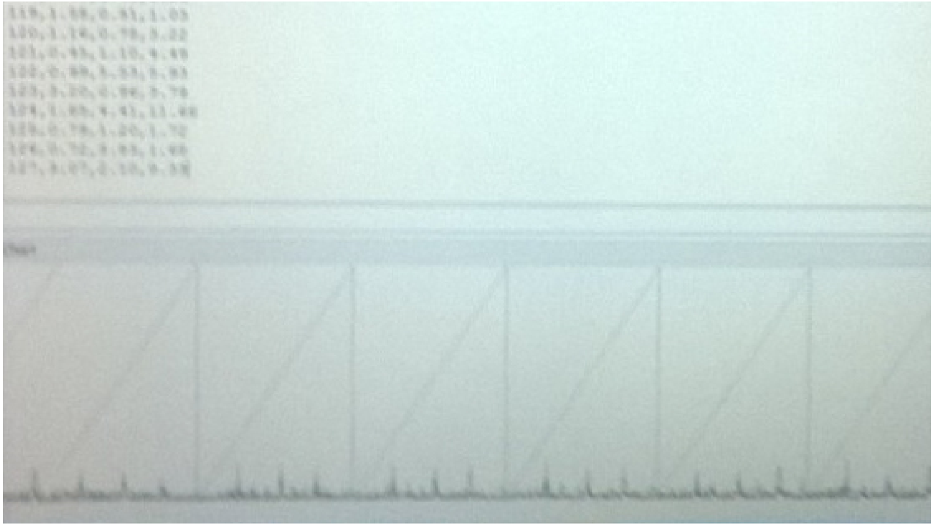

Figure B.7. *LH410 diesel engine idle*

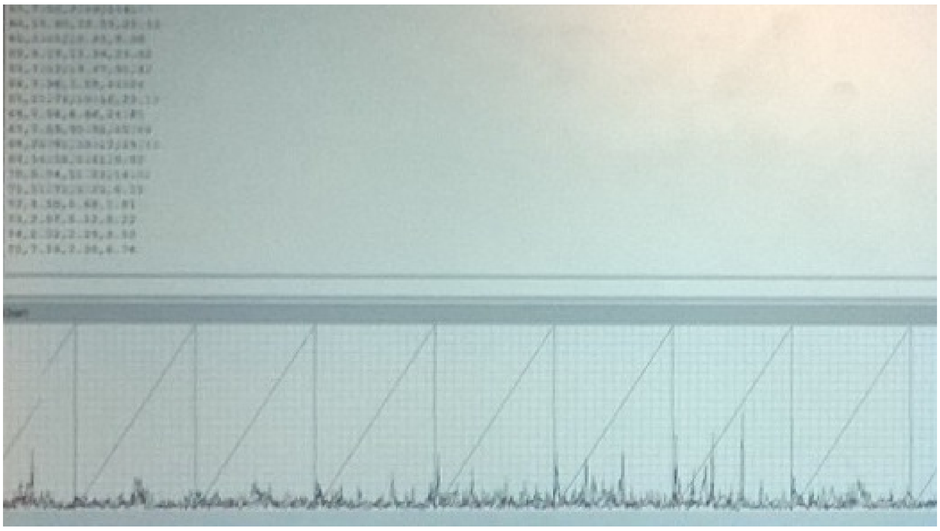

Figure B.8. *LH410 hydraulic (bucket) in use*

Figure B.9. *Test COSMOS on the LH410 loader*